Técnicas Avançadas para Análise Visual de Redes Temporais

Cláudio Douglas Gouveia Linhares

Universidade Federal de Uberlândia
Faculdade de Computação
Programa de Pós-Graduação em Ciência da Computação

Uberlândia
2020

Cláudio Douglas Gouveia Linhares

Técnicas Avançadas para Análise Visual de Redes Temporais

Tese de doutorado apresentada ao Programa de Pós-graduação da Faculdade de Computação da Universidade Federal de Uberlândia como parte dos requisitos para a obtenção do título de Doutor em Ciência da Computação.

Área de concentração: Ciência da Computação

Orientador: Bruno Augusto Nassif Travençolo
Coorientador: José Gustavo de Souza Paiva

Uberlândia
2020

Ficha Catalográfica Online do Sistema de Bibliotecas da UFU
com dados informados pelo(a) próprio(a) autor(a).

L755 Linhares, Cláudio Douglas Gouveia, 1990-
2020 Técnicas Avançadas para Análise Visual de Redes Temporais [recurso eletrônico] / Cláudio Douglas Gouveia Linhares. - 2020.

Orientador: Bruno Augusto Nassif Travençolo.
Coorientador: José Gustavo de Souza Paiva.
Tese (Doutorado) - Universidade Federal de Uberlândia, Pós-graduação em Ciência da Computação.
Modo de acesso: Internet.
Disponível em: http://doi.org/10.14393/ufu.te.2020.375
Inclui bibliografia.
Inclui ilustrações.

1. Computação. I. Travençolo, Bruno Augusto Nassif,1981-, (Orient.). II. Paiva, José Gustavo de Souza,1979-, (Coorient.). III. Universidade Federal de Uberlândia. Pós-graduação em Ciência da Computação. IV. Título.

CDU: 681.3

Bibliotecários responsáveis pela estrutura de acordo com o AACR2:
Gizele Cristine Nunes do Couto - CRB6/2091
Nelson Marcos Ferreira - CRB6/3074

Este trabalho é dedicado às crianças adultas que,
quando pequenas, sonharam em se tornar cientistas.

Acknowledgement

To my parents, Enivaldo and Elcione, and my brother, Hélder, by the love, incentive and unconditional support, because without them, nothing would be possible.

To my beautiful lifemate, Sara Fernandes, that always give me strength and support in all my decisions. Her dedication inspires me and it is very important to the conclusion of this thesis.

To my friends Jean Ponciano and Everton Lira that were a crucial part of all my academic courses, always being helpful and partners.

To the professors Bruno Travençolo, José Gustavo and Luis Rocha, by the guidance, support, and trust.

To all that were directly or indirectly a part of my formation, I appreciate it.

To the research funding agency CAPES for the scholarship granted to me.

"I would rather have questions that can't be answered than answers that can't be questioned."

(Richard Feynman)

Resumo

Redes temporais representam interações entre entidades do domínio com a informação adicional de quando essas comunicações ocorrem. A visualização de redes temporais tem um importante papel no reconhecimento de propriedades das redes que seriam difíceis de serem percebidas sem uma estratégia de visualização adequada. Devido à grande quantidade de informação nessas redes, mais atenção tem sido dada em relação a escalabilidade visual associada com visualizações produzidas, mas ainda representa um problema não resolvido e com falta de abordagens específicas. Neste trabalho são propostas novas estratégias para melhorar a visualização de redes temporais. Especificamente, é proposta uma técnica de ordenação de nós escalável para a visualização de redes temporais, chamada de *Community-based Node Ordering (CNO)*, que combina detecção de comunidade estática com técnicas de ordenação de nós, juntamente com uma taxonomia para categorizar os padrões das atividades. É apresentado também um método de visualização que permite a comparação entre dois algoritmos de detecção de comunidades para ajudar a decidir qual deles é melhor para a análise visual de comunidades. Também são abordados estratégias para a visualização de processos dinâmicos em redes, como espalhamento de rumores e doenças. Além disso, é conduzido um experimento com usuário com a definição de diferentes tarefas em redes temporais, a fim de identificar quais são as melhores formas de visualizar de acordo com diferentes tarefas. Por fim, é descrito o sistema *Dynamic Network Visualization (DyNetVis)*, mostrado suas especificações, requisitos, funcionalidades e impacto na área. Os experimentos foram gerados com análises quantitativas e qualitativas utilizando redes reais em diferentes contextos, para mostrar que as visualizações propostas e suas categorizações ajudaram na identificação de padrões que seriam difíceis de serem vistos sem o uso dessas técnicas de visualização.

Palavras-chave: Redes Temporais. Redes Dinâmicas. Comunidades. Reordenação de nós. *Massive Sequence View*. Poluição Visual. Escalabilidade Visual. Visualização da Informação. Processos Dinâmicos. DyNetVis.

Advanced Techniques for Visual Analysis of Temporal Networks

Cláudio Douglas Gouveia Linhares

Universidade Federal de Uberlândia
Faculdade de Computação
Programa de Pós-Graduação em Ciência da Computação

Uberlândia
2020

UNIVERSIDADE FEDERAL DE UBERLÂNDIA
Coordenação do Programa de Pós-Graduação em Ciência da Computação
Av. João Naves de Ávila, nº 2121, Bloco 1A, Sala 243 - Bairro Santa Mônica, Uberlândia-MG, CEP 38400-902
Telefone: (34) 3239-4470 - www.ppgco.facom.ufu.br - cpgfacom@ufu.br

ATA DE DEFESA - PÓS-GRADUAÇÃO

Programa de Pós-Graduação em:	Ciência da Computação				
Defesa de:	Tese, 13/2020, PPGCO				
Data:	12 de março de 2020	Hora de início:	**14h30min**	Hora de encerramento:	**17h00min**
Matrícula do Discente:	11613CCP007				
Nome do Discente:	Cláudio Douglas Gouveia Linhares				
Título do Trabalho:	Técnicas Escaláveis para Análise Visual de Redes Temporais				
Área de concentração:	Ciência da Computação				
Linha de pesquisa:	Ciência de Dados				
Projeto de Pesquisa de vinculação:	-				

Reuniu-se na sala 1B132, Bloco 1B, Campus Santa Mônica, da Universidade Federal de Uberlândia, a Banca Examinadora, designada pelo Colegiado do Programa de Pós-graduação em Ciência da Computação, assim composta: Professores Doutores: Renan Gonçalves Cattelan - FACOM/UFU, Murillo Guimarães Carneiro - FACOM/UFU, Filipi Nascimento Silva - IUNI/INDIANA UNIVERSITY , Francisco Aparecido Rodrigues - ICMC/USP, José Gustavo de Souza Paiva - FACOM/UFU (coorientador) e Bruno Augusto Nassif Travençolo - FACOM/UFU, orientador do candidato.

Ressalta-se que o Prof. Dr. Filipi Nascimento Silva, participou da defesa por meio de videoconferência desde a cidade de Bloomington, EUA e o Prof. Dr. Francisco Aparecido Rodrigues da cidade de São Carlos - SP . Os outros membros da banca e o aluno participaram *in loco*.

Iniciando os trabalhos o presidente da mesa, Prof. Dr. Bruno Augusto Nassif Travençolo, apresentou a Comissão Examinadora e o candidato, agradeceu a presença do público, e concedeu ao Discente a palavra para a exposição do seu trabalho. A duração da apresentação do Discente e o tempo de arguição e resposta foram conforme as normas do Programa.

A seguir o senhor presidente concedeu a palavra, pela ordem sucessivamente, aos examinadores, que passaram a arguir o candidato. Ultimada a arguição, que se desenvolveu dentro dos termos regimentais, a Banca, em sessão secreta, atribuiu o resultado final, considerando o candidato:

Aprovado

Esta defesa faz parte dos requisitos necessários à obtenção do título de Doutor.

O competente diploma será expedido após cumprimento dos demais requisitos, conforme as normas do Programa, a legislação pertinente e a regulamentação interna da UFU.

Nada mais havendo a tratar foram encerrados os trabalhos. Foi lavrada a presente ata que após lida e achada conforme foi assinada pela Banca Examinadora.

Documento assinado eletronicamente por **Bruno Augusto Nassif Travençolo**, **Professor(a) do Magistério Superior**, em 13/03/2020, às 09:27, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do Decreto nº 8.539, de 8 de outubro de 2015.

Documento assinado eletronicamente por **José Gustavo de Souza Paiva**, **Professor(a) do Magistério Superior**, em 13/03/2020, às 09:44, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do Decreto nº 8.539, de 8 de outubro de 2015.

Documento assinado eletronicamente por **Renan Gonçalves Cattelan**, **Professor(a) do Magistério Superior**, em 13/03/2020, às 20:32, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do Decreto nº 8.539, de 8 de outubro de 2015.

Documento assinado eletronicamente por **Filipi Nascimento Silva**, **Usuário Externo**, em 30/03/2020, às 14:41, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do Decreto nº 8.539, de 8 de outubro de 2015.

Documento assinado eletronicamente por **Murillo Guimarães Carneiro**, **Professor(a) do Magistério Superior**, em 01/04/2020, às 17:46, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do Decreto nº 8.539, de 8 de outubro de 2015.

Documento assinado eletronicamente por **Francisco Aparecido Rodrigues**, **Usuário Externo**, em 08/04/2020, às 10:20, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do Decreto nº 8.539, de 8 de outubro de 2015.

A autenticidade deste documento pode ser conferida no site https://www.sei.ufu.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0, informando o código verificador **1929198** e o código CRC **C4A1E4BF**.

Abstract

Temporal networks represent interactions among entities of a given domain with additional information about when such interactions occur. The visualization of temporal networks plays a key role in the recognition of properties that would be difficult to perceive without an adequate visualization strategy. Due to a large amount of information provided by these networks, more attention has been given to issues related to the visual scalability associated with the produced layouts, but this still represents an unsolved problem and lacks effective techniques. We propose in this thesis novel techniques to enhance the visualization of temporal networks. Specifically, a scalable node reordering technique for temporal network visualization, named *Community-based Node Ordering (CNO)*, combining static community detection with node reordering techniques, along with a taxonomy to categorize activity patterns. In addition, a visualization method that allows the comparison of two community detection algorithms is presented in order to decide which one is better for visual analysis of communities. Another contribution is the analysis of dynamic processes, as spreading rumors, diseases, applied in the visualization of temporal networks. Furthermore, we conducted a user experiment consisting of the application of different tasks in temporal networks, in order to find the relation of the layouts with the most appropriate tasks. Finally, the *Dynamic Network Visualization (DyNetVis)* system demonstrates the software specifications, examples, functionalities, and impact in the study field. We performed experiments with qualitative and quantitative analyses using real networks in several fields to show that the proposed layouts and categorization helped in the identification of patterns that would otherwise be difficult to see.

Keywords: Temporal Networks. Dynamic Networks. Network Community. Node Reordering. Massive Sequence View. Visual Clutter. Visual Scalability. Information Visualization. Dynamic Processes. DyNetVis.

List of Figures

List of Tables

Acronyms list

CNO Community-based Node Ordering

DyNetVis Dynamic Network Visualization System

Hi-Fi High Fidelity

HCI Human-Computer Interaction

InVS Public Health Surveillance Institute

Lo-Fi Low Fidelity

MSV Massive Sequence View

RFID Radio-Frequency Identification

RN Recurrent Neighbors

SI (Susceptible - Infected) model

SIS (Susceptible - Infected - Susceptible) model

SIR (Susceptible - Infected - Recovered) model

SIRS (Susceptible - Infected - Recovered - Susceptible) model

TAM Temporal Activity Map

Contents

Chapter 1

Introduction

Complex interactions between the components of a system may be mapped into a complex network by representing these components as nodes and by connecting them according to the relation they have in the respective system. This modeling helps to identify meaningful relations between these components at different scales and to understand how these interactions regulate the diffusion of information or other dynamic processes, such as epidemics, taking place on the network. Network science has been established as an important research area to study complex systems in various contexts and finds applications in disciplines as diverse as computer science, biology, linguistics and economics (COSTA et al., 2011). The popularity of network science as an analytic tool is partially explained by the increasingly availability of high quality data, for example, of contact networks between humans (collected through RFID sensors) and online social networks (web-crawlers), or communication and mobility networks, extracted from mobile phone call records or e-mail accounts (KEILA; SKILLICORN, 2005), that reveal non-trivial patterns of interactions between the components of the system.

Network studies have traditionally focused on the structural properties of networks. However, recent studies have shown that networks not only have structure but also evolve, with edges (or connections) and nodes appearing and disappearing over time. As a consequence, the times in which edges occur have been shown to be sometimes as important as the structure of connectivity of the network. In these temporal (dynamic) networks, time stamps are assigned to edges to indicate the moment of activity. There are several dimensions that can be studied, such as the geographical dimension, but we focused instead in the temporal dimension to represent connections over time, adding an extra level of complexity to network analysis since activation times typically do not follow regular behavior but bursts of activity, circadian cycles, causal activation, among other patterns. Social networks are an example of temporal networks that contain social relations (edges) among people (nodes) that share similar interests. In some situations, such as the analysis of social network evolution, the difficulty to perform this analysis increases according to the network size, becoming significantly challenging.

Information Visualization, an area dedicated to represent abstract data in visual representations, comes to improve these analyses of the network structure with different techniques for visual representation, and to help interpretation and reasoning in data through interactive processes (TERGAN; KELLER, 2005). There are several approaches for visualization of complex networks (LINHARES et al., 2019b; LINHARES et al., 2019a; ELZEN et al., 2013; ELZEN et al., 2014; ZHAO et al., 2018), however, it is still a challenging area in scenarios in which the data are large in space and/or time and have complex structures (BECK et al., 2016). In temporal networks, it is even more challenging finding an appropriate layout, i.e., an adequate way of displaying elements on the screen according to organizational principles and creating an interactive visualization (LINHARES et al., 2019b).

An example of visualization technique is the Massive Sequence View - MSV (or Temporal layout), with each line representing a different node and each column representing a different time, which is ideal to provide an analysis of the temporal network evolution through the vertically positioned edges in the layout (Fig. 1(a)). Moreover, as the position of the nodes influences the edges size, it also impacts the visualization and, consequently, the perception of the data patterns. There are some visualization techniques that use this layout in literature, most of them employing node reordering (Fig. 1(b)) (CORNELISSEN et al., 2007; ELZEN et al., 2014). Also, there are some edge sampling strategies, i.e., an algorithm chooses according to a defined rule in which edges are hidden or shown in the layout (Fig. 1(c)), which was also used in the MSV layout (ZHAO et al., 2018). Despite recent progress in the field, visual exploration of temporal networks remains a challenging research topic without effective solutions to deal with visual clutter, particularly in large networks, i.e., a network with thousands of nodes/edges (ELZEN et al., 2014; MI et al., 2016; WU et al., 2016). Visualizing these networks raises new challenges related to scalability issues.

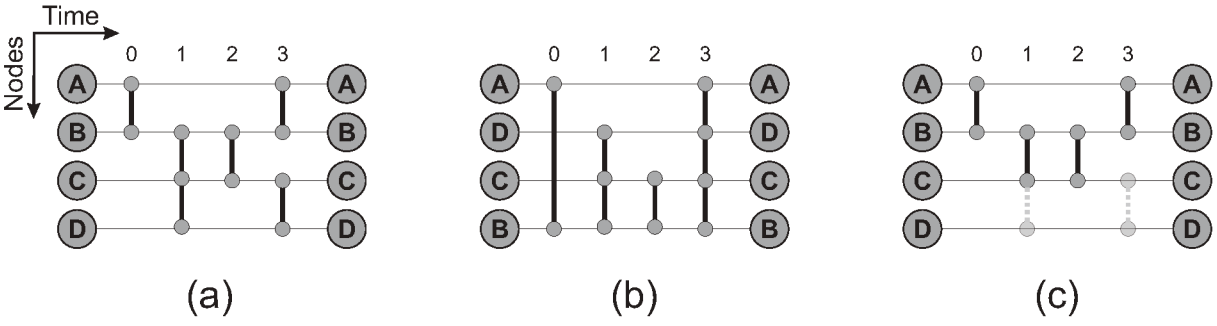

Figure 1 – Example of visualization technique to represent temporal networks called MSV layout. The x-axis represents the temporal dimension and the y-axis represents the nodes. (a) Original layout; (b) Using node ordering technique; (c) Using edge sampling technique.

Scalability, in our context, is related to how a strategy deal with small and large

networks. There are two types of scalability that may be considered: visual and computational. Computational scalability refers to the capacity of a system/algorithm to maintain low running time or computational resources while the input increases. Visual scalability indicates the capacity of a visualization technique to display large datasets improve the pattern recognition. In this thesis, there are solutions focused on visual scalability, i.e., visualization techniques that are specific for large networks (ABELLO; HAM; KRISHNAN, 2006; ELMQVIST et al., 2008; MI et al., 2016), but none of them provides solutions to networks that consider the temporal dimension. Only a few studies propose solutions with the focus on the visual scalability of temporal networks (BURCH, 2017; ROSVALL; BERGSTROM, 2010), which presents an open field for exploration of new visualization techniques.

Different visualization strategies suffer from scalability issues. Figure 2 represents a common visualization representation called node-link diagram, where the nodes are represented using circles and the edges are straight lines. Notice that in this static image the time is not represented, which can be solved adding an animation, highlighting nodes and edges that are active in the specific time stamps. However, even in the case of the static representation (node-link diagram) that is not dealing with the time of the connections, there are several challenges to deal with this illustrated network, that have no more than 2.000 nodes and edges. There are some popular techniques to enhance the node-link diagram for large networks, such as edge bundling (HOLTEN; WIJK, 2009) and

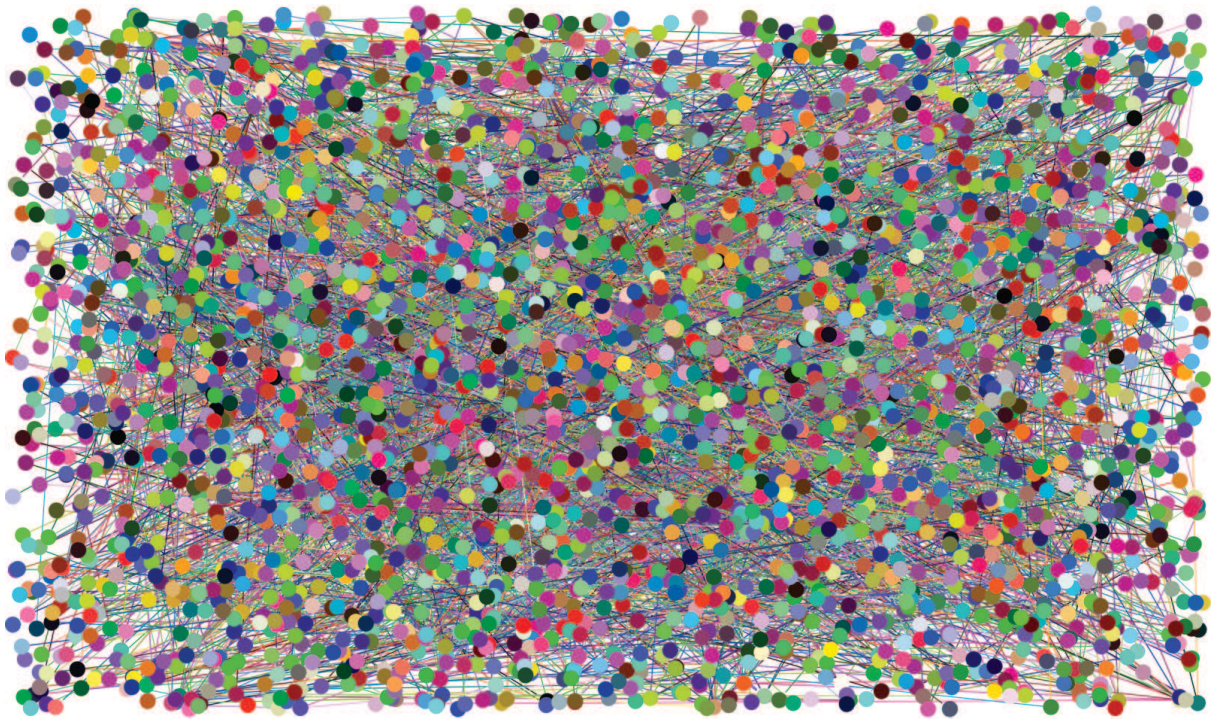

Figure 2 – Example of a problem in the network visualization using the node-link diagram. The network represented have no more than 2.000 nodes and edges.

force-directed layout (BRANDES, 2008). However, temporal networks are still a challenge for this representation, lacking more studies focused on temporal analyses.

An other commonly used method to deal with visual scalability for large networks is the community analysis. A community is defined as a group of nodes (entities of the network) that interact more among themselves than with nodes from other groups (FORTUNATO, 2010). The identification of these groups is important because they demonstrate how the network is organized and how the relationships among its components is structured, allowing to focus the analysis in specific regions of interest. In this way, it is possible, for example, to recognize groups of friends in social networks, proteins with similar functions, related diseases, etc. (FORTUNATO; HRIC, 2016). Although there are several methods in the literature that analyze community evolution over time, e.g., (Orman et al., 2014; ROSVALL; BERGSTROM, 2010; SALLABERRY; MUELDER; MA, 2013), this is not our focus. Instead, we use static community detection, i.e., considering the entire network at once (aggregated network), to approximate nodes that connect mostly among themselves, enhancing the analysis.

1.1 Motivation

Considering the wide applicability of temporal networks, several methods, layouts, and techniques were created to represent different tasks, however, finding appropriated techniques is still a challenge (BECK et al., 2014). One way to identify the best techniques for proper tasks is to conduct a user study, where, according to a given a task, the user is capable of choosing what is the best representation in a context. Considering the methods for network visualization, the most recent techniques have limited visual scalability (ELZEN et al., 2014; LINHARES et al., 2017b; BURCH, 2017; VEHLOW et al., 2015; ZHAO et al., 2018), becoming important further investigation in how to adapt or create novel techniques to allow a visualization for large networks.

1.2 Goals

The main research goal addressed in this thesis is to create and evaluate advanced techniques to visualize temporal networks, i.e., state-of-the-art techniques to enhance the analysis of temporal networks. The specific goals are:

- Aggregate the visual analysis in network community detection algorithms evaluation.
- Propose novel scalable techniques for temporal networks.
- Use visualization to analyze dynamic processes in temporal networks.

- Conduct a user experiment for quantitative and qualitative evaluation of visualization techniques for temporal networks.

- Combine our proposals and other techniques in a computational system with several interactive tools.

1.3 Hypothesis

To create and evaluate visualization techniques for temporal networks, we elaborate the following hypothesis:

Visualization techniques that receive the network community structure as input reduce visual clutter and thus enhance temporal network analysis.

1.4 Contributions

The contributions are related to the processes of creating advanced visualization techniques to temporal networks, and to the quantitative and qualitative analyses of these visualizations. Also, these techniques were applied to several real networks and analyzed through dynamic processes. The visualization tools used in this thesis were combined in a computational system. The contributions are described as follows.

1. A novel methodology for comparison of community detection algorithms using visual analysis, which helps the user to choose the more adequate method for a specific scenario.

2. A novel scalable node reordering technique for temporal networks, which improves the layout and thus facilitates the data analysis. Furthermore, a taxonomy is proposed to categorize different types of communities according to their activity patterns.

3. A user study for evaluation of four different layouts for temporal networks, in order to validate the relation between the user's tasks and layouts.

4. Case studies using the visualization of temporal networks and dynamic processes, allowing the user to understand the impact of the visualization in the studied network.

5. An open source computational system with several interactive tools, in order to combine and validate the efficiency of the proposed techniques.

1.5 Thesis organization

This thesis is organized as follows. Chapter 2 provides an overview of the area and describes the related works. Basic concepts of temporal network visualization, well-establish methods in the area, layout decisions and main techniques are discussed. Furthermore, it describes an overview of communities in the context of detection algorithms and visualization. Chapter 3 presents a methodology with a visualization technique for visual evaluation of the communities algorithms. Chapter 4 illustrates a novel scalable technique for temporal networks, along with a taxonomy to categorize communities, both evaluated using quantitative and real world case studies. Chapter 5 shows the use of visualization for dynamic processes in temporal networks. Chapter 6 provides a user experiment for several temporal layouts. Chapter 7 presents the developed system to prove the efficiency of the techniques proposed here. Finally, Chapter 8 shows the concluding remarks of this thesis.

Chapter 2

Fundamentals

This chapter describes concepts and proposals related to visualization of temporal networks, such as different layouts and techniques as well as their relationship with the network community structure.

2.1 Temporal Network visualization

An adequate network visualization is helpful to identify patterns and to support decision making. For temporal networks, the visual analysis demands adequate techniques that allow the identification of temporal patterns (BECK et al., 2014).

2.1.1 Basic concepts

The exploration of temporal networks is often infeasible via tabular analysis. Several visual strategies may be considered for this exploration, such as space-time cubes (BACH, 2016), circular methods (ELZEN et al., 2014), structural and temporal layouts (LINHARES et al., 2017b; BATTISTA et al., 1994). Although each of them presents advantages and disadvantages depending on the task, structural, matrix and temporal layouts are widely used to analyze the connections distribution.

The **Structural Layout**, also known as **node-link diagram**, is the conventional graph representation in which the entities (nodes) are spatially inserted on the layout with edges connecting them and is applied on temporal networks by adding timestamps on the edges, as illustrated in Fig. 3(a). This layout is most useful when the goal is to analyze the aggregated network, i.e., considering the entire network at once, facilitating global patterns identification. On the other hand, if the network has a lot of nodes/edges, the perception may be affected due to visual clutter. To attenuate this, several strategies may be employed on the structural layout to change the position of the nodes, such as force-based and circular algorithms (MI et al., 2016; BATTISTA et al., 1994; BASSETT et al., 2012; SIX; TOLLIS, 2006) and strategies for edge-bundling (HOLTEN; WIJK,

2009; LAMBERT; BOURQUI; AUBER, 2010; LHUILLIER; HURTER; TELEA, 2017). Nevertheless, even using animation graphs to represent temporal information, it is hard to maintain effectively the mental map preservation using structural layout (ARCHAMBAULT; PURCHASE, 2016).

The **Matrix Layout** (Fig. 3(b)) represents in each cell of a matrix the edge between the correspondent node line and node column, e.g., the colored cell (C, B) represents the edge between nodes C and B, for instance. Similarly to the structural layout, the matrix layout represents temporal information using animation, in which a cell becomes colored in a timestamp only if there is an edge between the nodes in such timestamp. The matrix layout is implemented as a square and symmetric matrix, i.e., both cells (A, B) and (B, A) represent the same connection. The matrix representation is more suitable for larger and denser networks than the structural layout, being preferred, for instance, for comparisons between networks (BECK et al., 2016; BEHRISCH et al., 2016). Besides, as time is represented by animation, the matrix suffers from the same issues as the structural layout.

The **Temporal Layout**, also known as **Massive Sequence View (MSV)** (CORNELISSEN et al., 2007), is a layout in which the vertical and horizontal axes represent the nodes and the time instants, respectively (Fig. 3(c)). In this representation, the nodes must have non-variant positions over time, i.e., once the nodes positions are defined, they cannot switch places over time in order to preserve the mental map. The objective of this layout is to analyze the network evolution through time by observing the edges vertically positioned and the amount of nodes connections in specific timestamps.

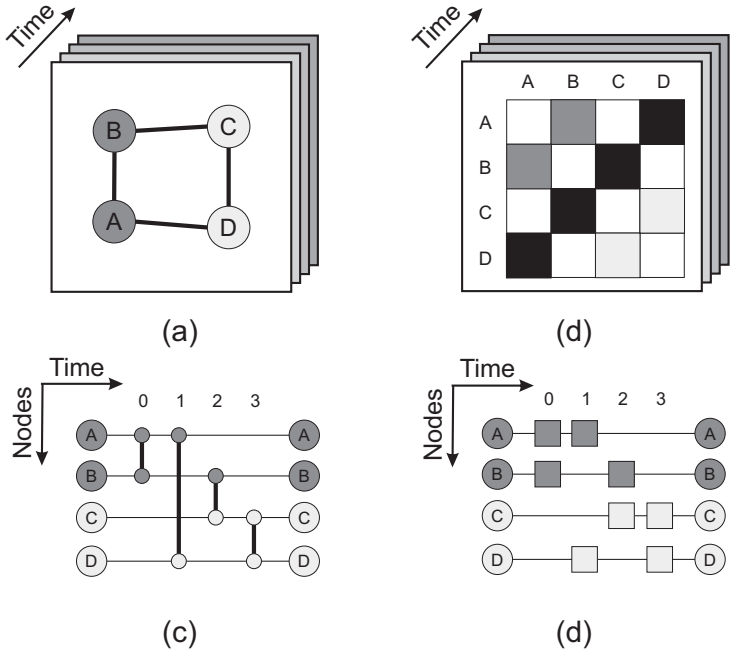

Figure 3 – Layouts to represent temporal networks: (a) Structural; (b) Matrix; (c) MSV; (d) TAM.

Node reordering is an important technique that tries to reduce visual clutter by defining node positioning. Node reordering algorithms are widely used in both static (e.g., visual matrices (BEHRISCH et al., 2016)) and temporal network visualization (e.g., MSV layout (ELZEN et al., 2014; LINHARES et al., 2017b)).

The layout **Temporal Activity Map (TAM)** is created to emphasize the node activity on MSV based on the number of connections that each node forms at each timestamp in respect to all connections in the network (Fig. 3(d)). This layout have rectangles instead of circles to represent the nodes. It also hides all edges. In the TAM layout, the node degree at each timestamp is computed for all nodes and the highest degree is taken as the maximum value of the activity range, that is used to map the color of nodes. The rectangular shape gives a better sense of continuity (WARE, 2013) and the absence of links decreases the number of elements in the layout, highlighting the node activity.

2.1.2 Visualization techniques for Temporal Networks

Visualization techniques are, in our context, a group of techniques that enhance the layouts, such as node ordering, edge sampling, and so on. Each layout can be enhanced by specific techniques, such as force-based algorithms, which uses a springs model to position the nodes in the structural layout according so that more connected nodes appear more central (ZHANG et al., 2005). Figure 4(a-b) present a node ordering process for the temporal (a) and matrix (b) layouts. Notice that depending on the chosen ordering, the layout perception is completely different. In the temporal layout, the size of the edges varies depending on the reordering approaches, which can overlap more frequently with each other, increasing the visual clutter. Another technique is the edge sampling, i.e.,

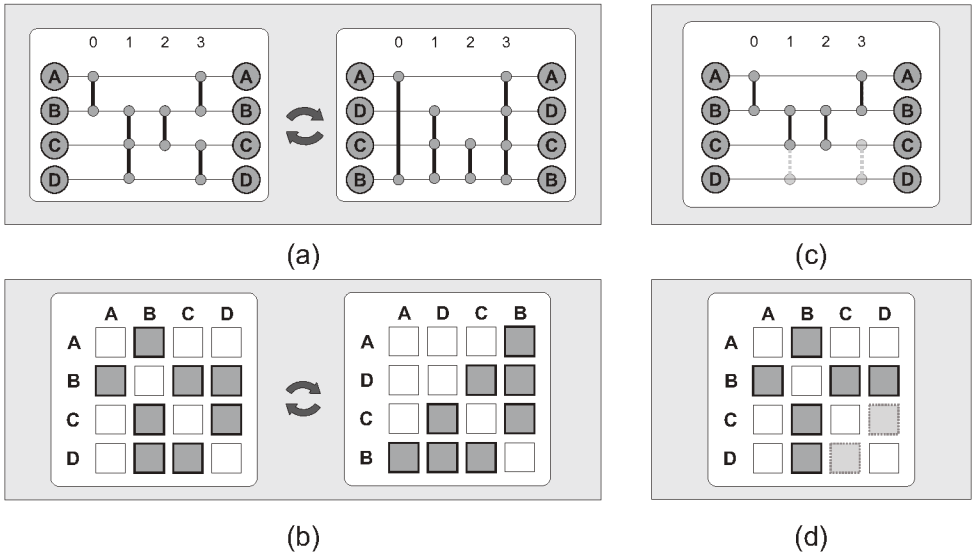

Figure 4 – Techniques for layouts to enhance visualization. (a) and (b) show a node ordering process for the temporal and matrix layouts, respectively. (c) represents an edge sampling process over the temporal layout.

the algorithm chooses according to some rule which edges are hidden or shown, changing the user perception depending on the resulting edges, as illustrated in the temporal and matrix layouts in Fig. 4(c-d).

Different node ordering were proposed previously, in order to better organize the layout and to reduce edge overlap. Some of these techniques are discussed as follows.

- **Appearance**: Nodes are sorted on ascendant order by the first time they have a connection on the network. In this view, it is possible to visualize how nodes are introduced in the network and obtain information as, for example, the time elapsed between the first and the last node appearance in the network. It also helps to verify if the first nodes that appeared on the network remain active over time or concentrate their activity on specific time periods (LINHARES et al., 2017b).

- **Lexicographic**: Nodes are sorted in ascendant order by a label, which varies according to the problem. If the nodes are labeled with numbers, it is used an ascendant numeric ordering. If the labels are letters or words, alphabetic order is employed (ELZEN et al., 2014).

- **Node degree**: Nodes are sorted in ascendant order by their accumulated degree. This strategy is useful to compare activities of nodes with a similar degree, as they are gathered in neighboring lines. For instance, two nodes with the same degree will be positioned in adjacent positions. These nodes may have significantly different connection patterns over time – the first can concentrate its connections in a specific time period whereas the other can interact along the entire time period (ELZEN et al., 2014).

- **Optimized MSV**: A hierarchical strategy that combines the minimization of both edge block overlap and standard deviation of the edges length. The first step of the algorithm is done using the simulated annealing optimization process, which tries to find the smaller combination of the edges length. An optimization in that result is made using the standard deviation of the edges length in the simulated annealing process. Additionally, the algorithm uses the minimization of the block overlapping, i.e., group of edges in the layout that are overlapping each other. Optimized MSV aims to reduce visual clutter using the optimization of the edges length and preventing unwanted visual attention using the block overlap minimization. Finally, Optimized MSV combines these two techniques using the Pareto optimal solution (ELZEN et al., 2014).

- **Recurrent Neighbors (RN)**: In this strategy, nodes whose connections are recurrent in time are maintained spatially closer. The idea is to approximate nodes with more connections with each other, highlighting node clusters and the connection distribution inside these clusters. This strategy minimizes the size of links in the

MSV layout, reduces the number of long edges and consequently the visual clutter. The reordering process is as follows: the node (reference node) with the highest accumulated degree of the network is positioned in the central row on the layout. The two nodes (K and L) with the highest number of connections with the reference node are then selected and placed on the rows above and below the reference node. A similar procedure is repeated for each of the nodes K and L, but now only one new node is selected for K and one for L. The new node M with the highest number of connections with node K (similar for node N in respect to node L) is positioned just above (below) it. This process is repeated until all nodes are exhausted. If there are unselected nodes in the network, the process starts over again (LINHARES et al., 2017b).

Even more elaborate methods, such as *Optimized MSV* and *Recurrent Neighbors*, have some critical issues. Due to a randomness of the simulated annealing process used in the *Optimized MSV*, it requires high computational time in order to find a good solution for a node reordering in a network. *Recurrent Neighbors* has a better computational time than in a good solution of *Optimized MSV*, since the simulated annealing process can take a while to find a good result. Figures 5, 6 illustrate both node ordering methods for an company network called Enron, with 151 nodes and 21.374 edges (KEILA; SKILLICORN, 2005). Important events in the network, such as the arrival of a new CEO, investigation and bankruptcy are indicated in both layouts. However, even in this case of a small network, is hard to observe the patterns due to the visual clutter, because both methods are not visually scalable, which is a problem for the analysis of large networks.

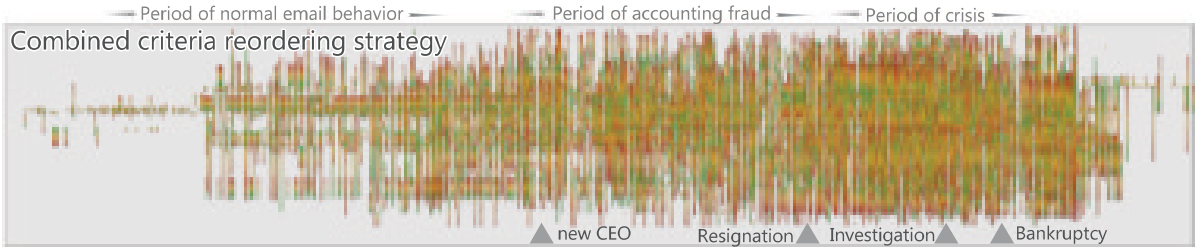

Figure 5 – Example of the node ordering technique called *Optimized MSV* for the Enron network. Adapted from (ELZEN et al., 2013) ©2013 IEEE.

Another technique on the MSV layout is to sample some edges according to some defined measure, as explained previously. In (ZHAO et al., 2018), it is created a sampling of the edges maintaining the edges distribution over time called *Edge Overlapping Degree* (EOD). The authors use the kernel density estimation to create an edge overlapping degree and characterize temporal features of node pairs, being able to choose which edges are hindering the analysis. The authors use the edge length and streaming process to enhance their sampling method, reducing the visual clutter in the MSV layout. They also

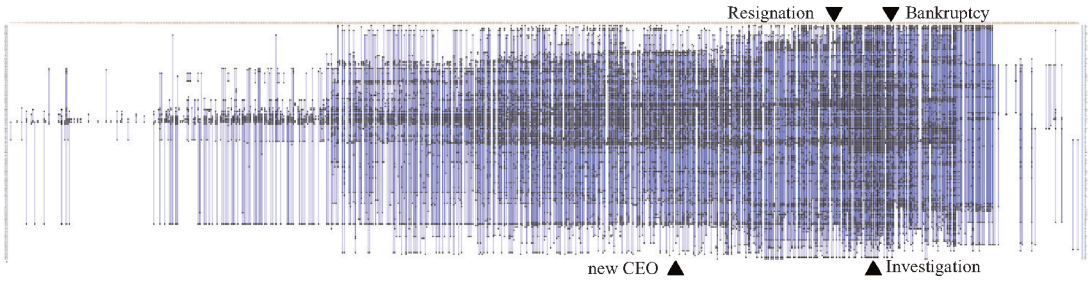

Figure 6 – Example of the node ordering technique called *Recurrent Neighbors* for the Enron network. Adapted from (LINHARES et al., 2017b) ©2017 ACM.

use the Enron network to evaluate the technique, as illustrated in Fig. 7. Similar to node ordering, in edge sampling is also hard to observe the patterns due to the visual clutter.

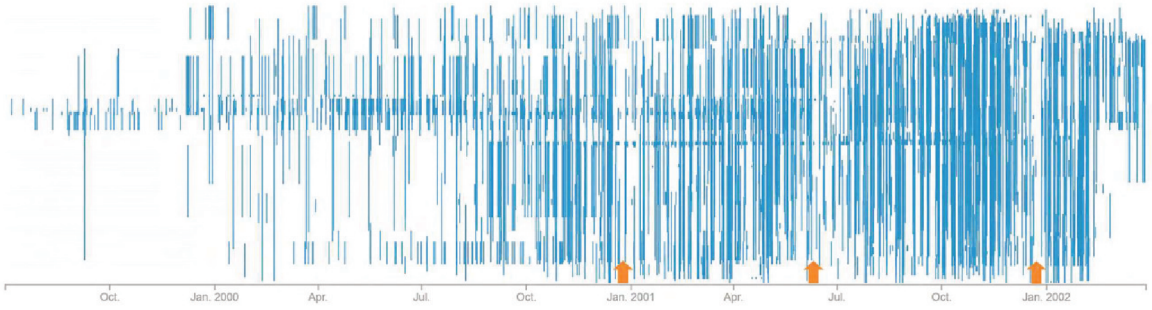

Figure 7 – Example of the edge sampling technique called *Edge Overlapping Degree* for the Enron network. Adapted from (ZHAO et al., 2018) ©2018 IEEE.

2.1.3 Interaction with the layout

Interaction is an important process for visualization techniques to improve the analysis by moving from global to local perspectives and vice-versa – as established by the Shneiderman's mantra *"overview first, zoom and filter, then details-on-demand"* (SHNEIDERMAN, 1996). One possible interaction is the choice of colors to represent nodes' or edges' attributes (Figure 8(b-c)). For example, one may associate nodes' colors to degree, community membership or other metadata. Another interaction is the possibility to select a specific region of interest in the layout, for example, all edges involving two or more nodes (Figure 8(d)). Analyzing nodes close to each other, one may select all nodes and edges in a particular group to focus the analysis. The user may also analyze edges at specific times (Figure 8(e)) or specific groups of nodes/edges (Figure 8(f)). Whenever a group of edges is selected, other components of the layout are dimmed to highlight the selected nodes and edges.

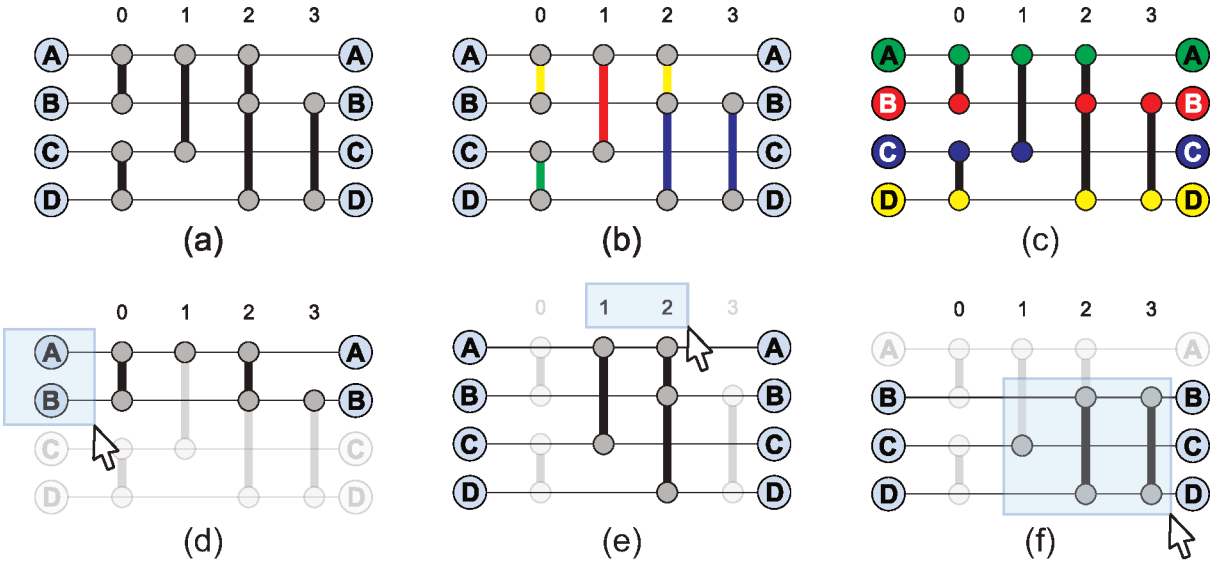

Figure 8 – Interacting with the MSV layout. (a) Original layout; (b) Using color to represent an edge attribute; (c) Using color to represent a node attribute; (d) Selection of all edges involving specific nodes; (e) Selection of all edges and nodes occurred in specific timestamps; (f) Selection of specific nodes and edges. Reprinted from (LINHARES et al., 2019b) ©2019 Elsevier.

2.1.4 Layout decisions

In order to enhance pattern identification and improve the experimental analysis, we made several layout decisions related to the construction of the structural and MSV layouts, such as the use of spherical nodes and straight edges, that were based on successfully implemented techniques used by Elzen et al. (ELZEN et al., 2016) and Linhares et al. (LINHARES et al., 2017b). Additionally, each of the layouts has its own particularities and thus requires specific decisions.

For better pattern identification and visual analysis in temporal networks, it is necessary to understand the concepts of Gestalt Laws, a set of laws that describes the way the human beings see patterns in a visual representation (WARE, 2013). The most relevant principles of the Gestalt Laws that may be applied in this thesis are closure, proximity, and similarity. The closure principle is described as *"a closed contour tends to be seen as an object"* (WARE, 2013). Fig. 9(a) shows an example of this principle in which the distinct nodes A and C are partially overlapped. Because of the closure of the contour in this figure, the brain visually separates the elements. In the same way, one can see the gray lines as a single block in Fig. 9(b). The proximity principle is interpreted as *"things that are close together are perceptually grouped together"* (WARE, 2013), as illustrated by the nodes A and C in Fig. 9(a) and by the gray blocks in Fig. 9(b). At last, the similarity principle is described as *"similar elements tend to be grouped together"* (WARE, 2013), as illustrated by the nodes B and E in Fig. 9(a). Even these nodes being spatially distant,

the brain tends to distinguish them from the other nodes due to the color in common. In Fig. 9(b), the lines are also visually segregated by the color difference.

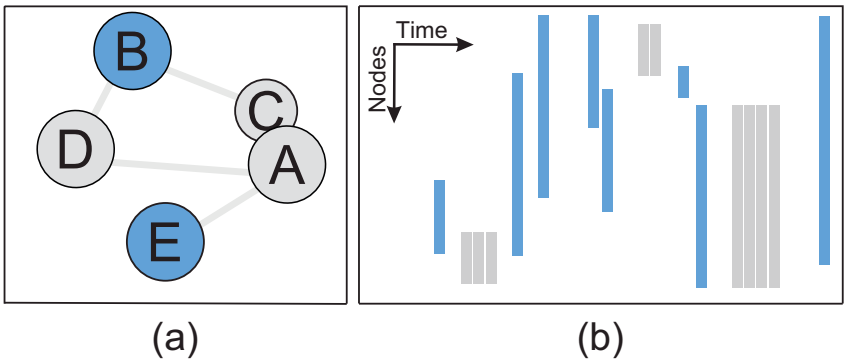

Figure 9 – Structural and MSV layouts on complex networks visualization. Distinct examples using Gestalt Laws applied to graphs: (a) structural layout representation, with different colors and overlap in the nodes; (b) MSV layout with different colors, positions and heights on edges. Adapted from (LINHARES et al., 2019a) ©2019 Springer Nature Switzerland AG.

Furthermore, in the structural layout, we decided to use redundant coding (WARE, 2013) by combining the edge thickness with the gray-scale color in a way that the more connections a pair of nodes have between themselves, the darker and thicker is the edge that connects them. We also associate different colors to represent different characteristics of nodes and/or edges. This decision, makes it easier, for instance, to visualize the relations between nodes within the same group. It is important to highlight that the association of colors to nodes and edges is only possible for labeled networks. Finally, the Gestalt's closure law is naturally considered on the MSV layout as the edges disposal visually creates dense regions, perceived as blocks of connections.

2.1.5 Resolution on Temporal Networks

One strategy that tries to improve the MSV layout on large networks is to change the original scale of time (temporal resolution) of the network by grouping consecutive timestamps (LINHARES et al., 2017b). Nevertheless, only the resolution change may result in a loss of information and do not entirely solve the visual scalability problem, since the order of the nodes may affect the size of the remaining edges. This procedure rearranges the data in the temporal dimension, grouping connections from subsequent timestamps. Equation 1 represents the change in the temporal resolution, in which t'_{new} is the new timestamp, t_{ori} is the timestamp in the original resolution, t_{s} is the first timestamp and δ is the resolution factor, i.e., the scale in which the time will be modified. For instance, if $\delta = 2$, the new resolution will be double of the original and the network

will have half of the timestamps when compared to the original.

$$t'_{\text{new}} = \left\lfloor \frac{t_{\text{ori}} - t_{\text{s}}}{\delta} \right\rfloor \delta + t_{\text{s}} \tag{1}$$

Figure 10 shows an example of the MSV layout before and after change in the temporal resolution, using $\delta = 2$. Node connections with different timestamps are grouped together in a single time. Changing the resolution of dynamic networks may facilitate the analysis when the network is temporally sparse, favoring the identification of patterns that can be difficult to identify using the original resolution.

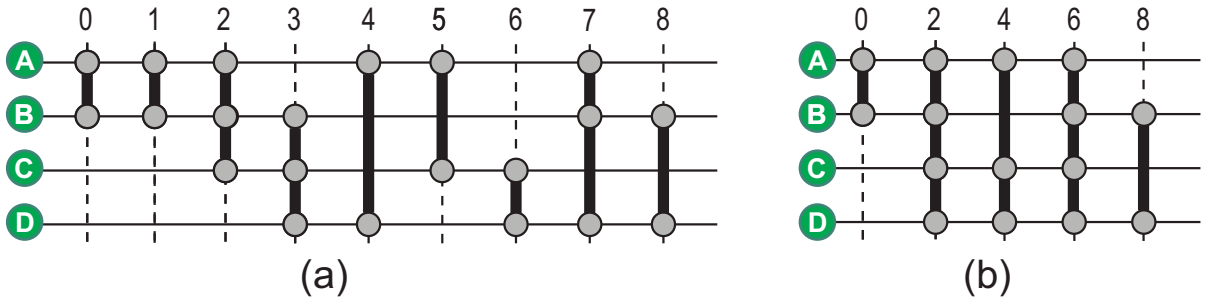

Figure 10 – Example of temporal resolution in the MSV layout. (a) Original resolution. (b) Change of resolution ($\delta = 2$). For example, nodes at timestamps 2 and 3 in the original version are grouped in time 2 in the new resolution.

It is important to notice that, when analyzing temporal networks, one should be aware of potential biases generated by sampling design. In particular, the temporal resolution and the period of observation (e.g., 1 day or 1 week) may affect the visual information. But this is not a visualization problem only since it is an intrinsic characteristic of temporal networks (ROCHA; MASUDA; HOLME, 2017). The MSV layout can accommodate such limitations but naturally the visual information will be different according to different choices of temporal parameters.

2.2 Communities in Temporal Networks

This section describes the definition of community for temporal networks, along with two of the most popular network community detection algorithms available in the literature (FORTUNATO; HRIC, 2016): Infomap (ROSVALL; BERGSTROM, 2008) and Louvain (BLONDEL et al., 2008). It also reviews measures to evaluate community detection algorithms and how Information Visualization strategies can help in this context.

One important property of temporal networks is the community formation (LANCICHINETTI; FORTUNATO, 2009). However, there is no definition for the term "community" that is universally accepted, since it depends on the application being used (FORTUNATO, 2010). Despite this, the following definition is used in our context: a community is

a group of nodes that have more internal edges than external, i.e., the nodes in a community have more connections between themselves than to nodes in other groups (FORTUNATO; HRIC, 2016; FORTUNATO, 2010). One can relate the communities as clusters in the network, identifying, for instance, groups of friends in social networks, proteins with the same function, related diseases, among others (FORTUNATO; HRIC, 2016). Indeed, there are several traditional clustering methods, specially adapted to deal with community detection (WANG; STREET, 2014; GIALAMPOUKIDIS et al., 2016).

2.2.1 Community detection algorithms

In networks, information is stored in the structure of connections between nodes. There are several algorithms to extract information from this structure, as for example hierarchical relations or groups of nodes with certain properties. Figure 11 shows a sample network (drawn as a node-link diagram (BATTISTA et al., 1994)) with 12 nodes and 3 communities (represented by yellow, green or blue nodes). Network communities may represent different features of the network as for example, a city or neighborhood (FORTUNATO; BARTHLEMY, 2007), a class of high-school students (MASTRANDREA; FOURNET; BARRAT, 2015), or specific population such as sex-workers (ROCHA; LILJEROS; HOLME, 2011).

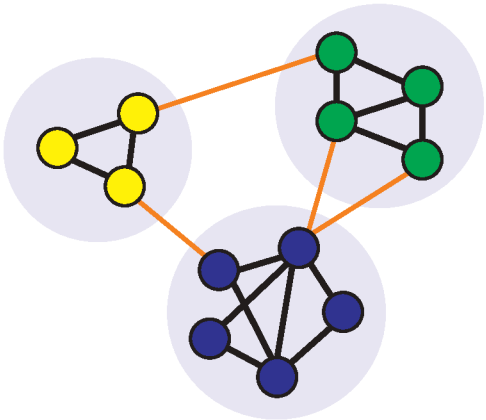

Figure 11 – Network represented by a node-link diagram with three communities where each community is represented by yellow, green or blue nodes. A network community is defined as a group of nodes that interact more often between themselves than with nodes from other groups. Reprinted from (LINHARES et al., 2020) ©2020 Springer Nature.

The detection of network communities is an important but also complex computational task (FORTUNATO; HRIC, 2016). Network community detection is similar to *network clustering* (FORTUNATO; HRIC, 2016) and thus traditional clustering algorithms that can be adapted to detect communities in networks (WANG; STREET, 2014; GIALAMPOUKIDIS et al., 2016). There are indeed several algorithms in the literature trying to

solve this problem using different approaches (ROSVALL et al., 2019; DRIF; BOUKERRAM, 2014; ORMAN; LABATUT; CHERIFI, 2012). Lancichinetti and Fortunato (LANCICHINETTI; FORTUNATO, 2009) evaluated 12 community detection algorithms considering several types of networks (i.e., undirected/directed, unweighted/weighted and random networks) and found that Infomap (ROSVALL; BERGSTROM, 2008) and Louvain (BLONDEL et al., 2008) have the best performance among the most popular algorithms. Follow up studies have also recommended both Infomap and Louvain as efficient solutions due to their fast response time and quality of results (MOTHE; MKHITARYAN; HAROUTUNIAN, 2017; ORMAN; LABATUT; CHERIFI, 2012). Both algorithms are $\mathcal{O}(m)$, where m is the number of edges in the network, making them suitable for the analysis of large networks with thousands of nodes and edges (FORTUNATO, 2010).

Infomap and Louvain algorithms are based on different principles to detect the network communities (ROSVALL et al., 2019; DRIF; BOUKERRAM, 2014; ORMAN; LABATUT; CHERIFI, 2012). Infomap represents a class of algorithms based on dynamic processes, in particular, a random walk process, used to identify coarse-grained descriptions of the network (ROSVALL et al., 2019). Since Infomap is based on random walks, it also belongs to a class of stochastic community detection algorithms (DRIF; BOUKERRAM, 2014). The algorithm essentially measures the quality of a community by using the path length of random walks (i.e., flow) or the time the random walker spends within a group of nodes. The network structure is then represented by a Huffman code and the optimal community structure is given by minimizing the amount of information required to represent the random walk. In other words, the algorithm detects communities by measuring groups of nodes in which the random walker gets trapped more often. The Louvain method, on the other hand, is based on modularity optimization (DRIF; BOUKERRAM, 2014). Modularity measures the strength of group structure in comparison to a random network with the same number of nodes and edges (see a formal definition in the next section). Louvain is a greedy, hierarchical and agglomerative algorithm where each node is initially considered a single community. To find the best partition of the network, the algorithm tries to heuristically merge (or group) these nodes in communities such that modularity is maximized.

2.2.2 Statistical community detection evaluation

A number of studies have analyzed the performance of community detection algorithms using statistical methods (FORTUNATO; HRIC, 2016; YANG; ALGESHEIMER; TESSONE, 2016; YIN et al., 2015; ORMAN; LABATUT; CHERIFI, 2012; MOTHE; MKHITARYAN; HAROUTUNIAN, 2017). There are several measures to analyse communities, such as Normalized Mutual Information (NMI), Triangle Participation Ration (TPR), Adjusted Rand index, among others (MOTHE; MKHITARYAN; HAROUTUNIAN, 2017). However, we focused on four common and highly used statistical measures:

Precision (Eq. 2), Recall (Eq. 3), F-Measure (Eq. 4) and the Modularity (Eq. 5) (YANG; ALGESHEIMER; TESSONE, 2016; YIN et al., 2015; NEWMAN, 2016).

Precision (also known as Purity) represents the ratio between the number of correctly detected communities and the number of detected communities $|R_i|$. Its varies from 0 (worst) to 1 (best).

$$Precision(R,T) = \frac{\sum_{i=1}^{p} \max_j(|R_i \cap T_j)|/|R_i|)}{p} \tag{2}$$

where $T = \{T_1, T_2, T_3, ..., T_q\}$ is the set of known ("real") communities in the network (e.g., a class of students or people living in the same city) and $R = \{R_1, R_2, R_3, ..., R_p\}$ is the set of detected communities using some clustering algorithm. Precision is not a good measure to evaluate community detection by itself. For example, considering each node as a single community will produce a maximum value of Precision (YIN et al., 2015).

Recall (also known as Collocation) is a value between 0 (worst) and 1 (best) that represents the ratio between the number of correctly detected communities and the number of known communities $|T_j|$.

$$Recall(R,T) = \frac{\sum_{j=1}^{q} \max_i(|R_i \cap T_j)|/|T_j|)}{q} \tag{3}$$

It is also not a good measure to evaluate community detection by itself. For instance, a single community including all nodes will produce a maximum Recall value (YIN et al., 2015).

F-Measure provides a balance between Precision and Recall. It varies between 0 (worst) and 1 (best) and it is computed as the harmonic mean between the Precision and Recall measures. If the detected communities are similar to the known communities, the F-Measure is close to 1 (YIN et al., 2015).

$$F\text{-}Measure(R,T) = \frac{2 \cdot Precision(R,T) \cdot Recall(R,T)}{Precision(R,T) + Recall(R,T)} \tag{4}$$

While the evaluation of F-Measure requires a ground-truth (known communities), Modularity can be computed for any network, even without a ground-truth associated. The Modularity value varies from -1 (worst) to 1 (best), and it is used to indicate the quality (or strength) of a particular partition in the network.

$$Modularity = \frac{1}{2m} \sum_{i,j} \left(A_{ij} - \frac{k_i k_j}{2m} \right) \delta(c_i, c_j) \tag{5}$$

where m is the number of edges in the network and A_{ij} is the value of the adjacency matrix A representing if there is an edge $A_{ij} = 1$ (or not $A_{ij} = 0$) between nodes i and j. The variable k_i is the degree of i (i.e., number of edges of i) and δ is the Kronecker delta (i.e., $\delta(c_i, c_j) = 1$ if i and j are in the same community and $\delta(c_i, c_j) = 0$ otherwise). Modularity values between 0.3 and 0.7 usually indicate good partitioning or strong community structure (YIN et al., 2015).

2.2.3 Visual analysis of network community structures

There are several studies that employed visualization techniques to improve the perception of structural patterns using with network communities (WANG et al., 2006; CRAMPES; PLANTIÉ, 2014; ROSVALL; BERGSTROM, 2010; TANAHASHI; MA, 2012; VEHLOW et al., 2015; RAJPOOT et al., 2015; TRAUD et al., 2009; DUNNE; SHNEIDERMAN, 2013; LINHARES et al., 2019b). Node-link diagrams to visualize the communities in the network is a common visualization approach (TRAUD et al., 2009; DUNNE; SHNEIDERMAN, 2013), however, instead of visualizing all nodes of each community independently, the nodes can be grouped into a single "super" node (Fig. 12(a)). In this case, the size of the node can represent the size of the community, i.e., the number of nodes inside a community, and the colors can represent different communities. In Ref. (RAJPOOT et al., 2015), the authors created a matrix-based layout to visualize communities, with colors representing communities and with the x-axis and y-axis representing the nodes (Fig. 12(b)). Rosvall and Bergstrom (ROSVALL; BERGSTROM, 2010) presented an alluvial diagram for the representation of communities that change over time, visually highlighting the merge/split and grow/shrink dynamics of these communities (Fig. 12(c)). The width of the horizontal boxes can represent the size of the communities but typically represents the variation in the cumulative flow (given by summing the PageRank of individual nodes) within the communities. Their study focuses on the evolution of the community and is not used to compare community detection algorithms.

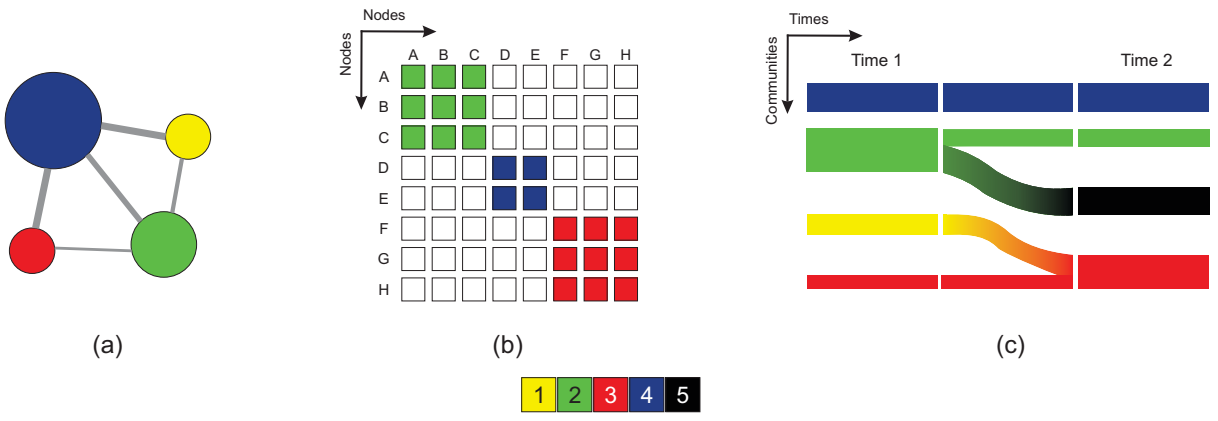

Figure 12 – Different strategies to visualize network communities. (a) Node-link diagram in which all community nodes are grouped and represented as a single node; (b) Matrix representation; (c) Alluvial diagram. Reprinted from (LINHARES et al., 2020) ©2020 Springer Nature.

Chapter 3

Visual Analysis for Community Detection Algorithms

This chapter introduces a methodology to analyze the performance of community detection algorithms using network visualization. We assess the methodology using two widely adopted community detection algorithms: Infomap and Louvain. We apply both algorithms to four real-world networks with a variety of characteristics to demonstrate the usefulness and generality of the methodology. We discuss the performance of these algorithms and show how the user may use statistical and visual analytics to identify the most appropriate network community detection algorithm for a certain network analysis task. This chapter is based on the article (LINHARES et al., 2020). The methodology that will be presented in this chapter is a result of a collaboration with another PhD candidate from our research group. We both contributed equally. This is a post-peer-review, pre-copyedit version of an article published in Multimedia Tools and Applications. The final authenticated version is available online at: <http://dx.doi.org/10.1007/s11042-020-08700-4>.

3.1 Community detection evaluation

A quantitative analysis using these or other measures may represent a "black-box", since the user just receives a numeric output, without interacting with the data. The user may have difficulty in understanding both the numeric result and its relation with the network under analysis, not being able to see how the nodes are distributed into communities, the community sizes and other relevant patterns. The choice of which community detection algorithm best represents a network behavior may thus be impaired. The application of visualization techniques includes the user in the analysis, making it more intuitive and facilitating the interpretation of the network data. The user also becomes able to choose which detection method is more adequate for a particular context. The integration of both statistical and visualization strategies further improves the user analysis (PERER; SHNEIDERMAN, 2008). Although there are several studies that pro-

pose community structure visualization (as illustrated in Fig. 12), none of them analyzed different community detection algorithms to choose the most appropriated.

3.2 Methodology

The methodology encompasses a set of steps to evaluate community detection algorithms to help in the decision of which one is the most appropriate in a specific scenario. Figure 13 illustrates the methodology workflow. The initial step (step 0) is to choose two community detection algorithms and the network data to study. Then, the methodology is divided into three main steps as follows:

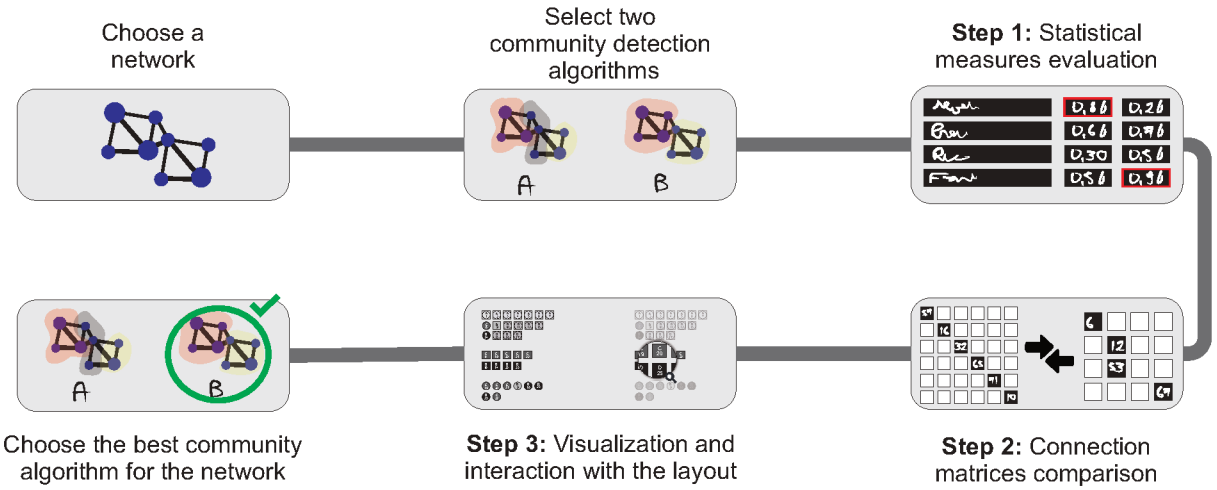

Figure 13 – Diagram representing the proposed methodology. First, a network and two community detection algorithms are chosen. Then, three main steps are evaluated: statistical measures, connection matrices, and visualization. After following all the steps, the most appropriate community detection algorithm is chosen. Reprinted from (LINHARES et al., 2020) ©2020 Springer Nature.

Step 1 - *Statistical measures evaluation:* This step consists in analyzing the quality of the community detection algorithms by employing the most appropriate statistical measures for community evaluation. This is important to evaluate the accuracy of the detection and node connectivity inside a community in relation to the other nodes. It is possible to use Modularity, Precision, Recall, and F-Measure in this step (described in section 2.2) since they measure different characteristics of the detected communities. For these measures, higher values (close to 1) indicate better performance.

Step 2 - *Connection matrices comparison:* In this step, the connection matrices for each community detection algorithm are compared to evaluate the coherence between the number of connections inside each community and how this community

interacts with other communities. In the connection matrices, each column and row represents a different detected community and the value in each cell represents the number of connections among the communities represented by the cell indices. The optimal result for these matrices is higher values in the main diagonal, meaning that nodes from a specific community are connecting more between themselves than to nodes from other communities. This pattern is represented in the matrix by coloring in black, for each column, the cell with the highest number of connections. Matrices with a black diagonal is a sign of good community detection. The matrix dimension also represents the number of communities that were detected by a specific community detection algorithm.

Step 3 - *Visualization and interaction with the layout:* In the last step, the user visualizes each detection algorithm and interact with the layout to evaluate the distribution of nodes in communities and the relation of these nodes with the nodes' labels (i.e., the node metadata). Using the visualization proposed next in Section 3.2.1 and the layout interaction possibilities presented in Section 2.1.2, the user can perform a visual analysis of the two community detection algorithms and better understand which are the similarities and differences between them. This step allows the identification of global patterns, e.g., the difference among the community distribution of nodes when comparing both algorithms; and local patterns, e.g., the frequency of a specific label among the nodes of a community (mixed labels or a predominant one). The user can also identify patterns and trends that would be otherwise hard to identify, thus improving the user experience.

The most appropriated community detection algorithm for a specific network is the one with the best performance in most steps.

3.2.1 Layout decisions

The layout to visualize the network communities is designed such that each row represents a community and nodes belonging to a specific community are placed one next to the other in the same row (Fig. 14(a)). To distinguish the community detection algorithms, a node from the first algorithm (Alg1) is represented by a square and nodes from the second algorithm (Alg2) are drawn using circles. This layout is used to visualize the communities detected by the two algorithms in three ways: (i) equivalent communities; (ii) non-equivalent communities obtained by Alg1; (iii) non-equivalent communities obtained by Alg2. Equivalent communities are defined as communities detected by both algorithms and sharing a large enough number of nodes. In other words, a community detected using algorithm Alg1 and another community detected using Alg2 are equivalent if the percentage of nodes appearing in both communities is higher than a pre-defined threshold T_{Eq}. In this case, all nodes from both communities are shown together in the

same row. If a node belongs to both communities, it is merged in one single representation using both a circle and a square with transparency; otherwise it is shown separated using a square if only in the community detected by Alg1 or using a circle if only in the community detected by Alg2 (Fig. 14(a)). In the case that non-equivalent communities are shown, the same visual strategy is employed to show communities detected by Alg1 (Fig. 14(b)) and by Alg2 (Fig. 14(c)), excluding equivalent communities.

The choice of the threshold T_{Eq} significantly influences the layout and consequently the visual analysis and decision making. It is expected that the number of equivalent communities increases as T_{Eq} decreases, since all equivalent communities for $T_{Eq} = n\%$ are also equivalent for $T_{Eq} = m\%$, for any $m \leq n$. Therefore, the number of equivalent communities is maximum if $T_{Eq} = 0\%$. A small T_{Eq} considers as equivalent even those communities with too few common nodes, that may not represent a real equivalence. On the other hand, higher T_{Eq} will largely restrict the occurrence of equivalent communities, possibly impairing the analysis of real-world networks. In practice, T_{Eq} may be chosen through an exploratory analysis considering the trade-off between its value and the quality and quantity of equivalent communities.

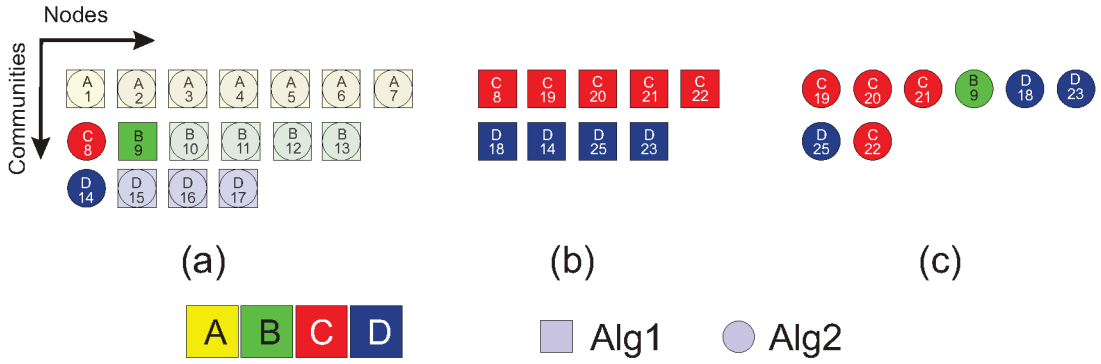

Figure 14 – The visual analysis to evaluate two community detection algorithms. (a) Equivalent communities; (b) Non-equivalent communities from Alg1; (c) Non-equivalent communities from Alg2. The color represents the node label, that is also written inside each node (redundant coding). Reprinted from (LINHARES et al., 2020) ©2020 Springer Nature.

Another important element in this visualization is the representation of a node label, which is made by associating a color to a node. To enhance this and respect the redundant color-coding (WARE, 2013), the node label is also associated with textual representation inside each node. This redundancy facilitates local analysis of nodes, e.g., identifying in which community a specific node was inserted by an algorithm. This association also makes possible to visualize the distribution of labels inside the communities and identify the prevalence of features in specific communities. For example, Fig. 14(b) shows that the sizes of the two communities are similar and that nodes in each community have the same labels (i.e., the communities captured the clustering of specific features). In contrast,

Fig. 14(c) shows that there are differences in the sizes of the communities and labels are mixed within communities without a clear predominance of one or another feature.

3.2.2 Interacting with the layout

To improve the visual analysis, interaction possibilities were included in the methodology to allow the user to follow the visualization mantra *"overview first, zoom and filter, then details-on-demand"* (SHNEIDERMAN, 1996), allowing the analysis from global to local perspectives and vice-versa (Fig. 15). The user can zoom and pan to freely navigate in the layout (Fig. 15(a)), enhancing the readability of the node textual information (e.g., node id) and nodes/communities, depending on the size of the network. Furthermore, the user can select specific nodes (Fig. 15(b)), improving the analysis of such nodes and communities, depending on the task. In this case, the selected nodes are highlighted on the layout, and the non-selected nodes are hidden using grayscale and transparency. With these interactions, the user can see the distribution of nodes on the detected communities and focus on regions of interest.

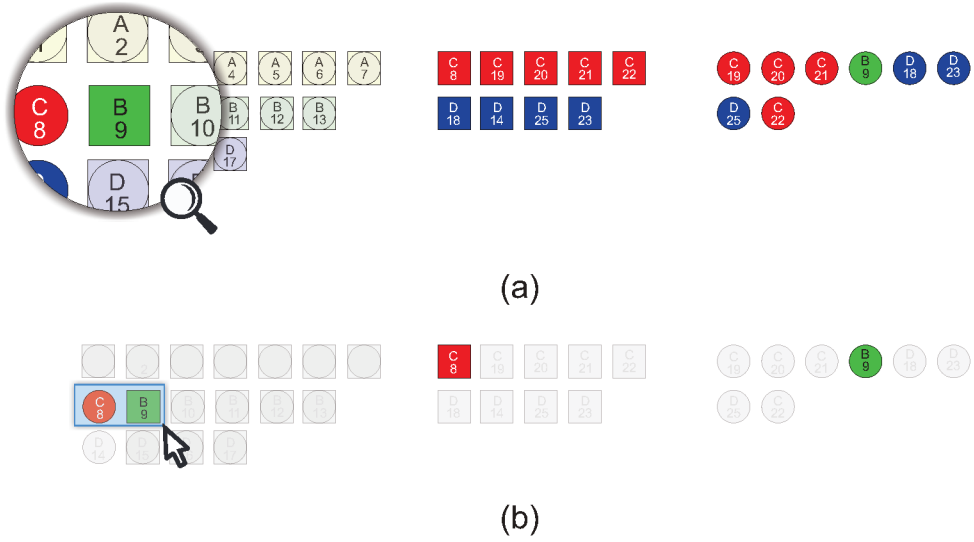

Figure 15 – Possible interactions to allow the user to observe the distribution of nodes and/or focus on a specific region of interest, thus improving the decision making of choosing the most appropriate method. (a) Zooming and Panning; (b) Nodes selection. In (a) and (b), the first column represents the Equivalent communities, the second column represents Non-equivalent communities from Alg1, and the third column represents the Non-equivalent communities from Alg 2. Reprinted from (LINHARES et al., 2020) ©2020 Springer Nature.

3.3 Experimental results

This section presents four case studies where we apply the methodology introduced in Section 3.2 to study real-world social network data in different contexts, one primary

(STEHLÉ et al., 2011) and one high school (MASTRANDREA; FOURNET; BARRAT, 2015), one university hospital (VANHEMS et al., 2013) and a public health surveillance institute (GÉNOIS et al., 2015). Each network corresponds to face-to-face proximity interactions (between 1 to 1.5 meters) between individuals detected using RFID sensors in each context. Contacts are recorded every 20 seconds (maximum temporal resolution) and the contact between these individuals is interrupted if no information is exchanged between the sensors (CATTUTO et al., 2010; VANHEMS et al., 2013). Although the methodology is general, we will use the Infomap and Louvain community detection algorithms (see Section 2.2.1) because of their popularity in the field of network data analysis.

Given the stochastic nature of both algorithms, the communities vary slightly for each run of the algorithm. For simplicity, we thus take the best result out of 10 realizations of each algorithm. For the Louvain method, the best result is given by the highest modularity and for Infomap it corresponds to the best hierarchical partition (shortest description length). We define that two communities are considered "equivalent" if they have at least $T_{Eq} = 70\%$ nodes in common. The effect of T_{Eq} on the visual layout is discussed in Section 3.2.1.

3.3.1 *Primary school* case

The first network corresponds to interactions of students and teachers in a primary school located in Lyon, France (GEMMETTO; BARRAT; CATTUTO, 2014; STEHLÉ et al., 2011). The data were collected between the 1st and 2nd of October 2009 and the network has 242 nodes and 125,773 edges. Students are enrolled from the first to the fifth grade and each grade is divided into two classes (A and B) with a fixed teacher in each class. In this network, most of the edges appear between students of the same class (STEHLÉ et al., 2011). We thus associate a unique community to each class to define the "true" community to evaluate the Infomap and Louvain algorithms. Since the teacher responsible for each class is not identified in the data, they are removed from the analysis.

To evaluate the network according to methodology Step 1, Table 1 shows the results of Precision, Recall, F-Measure and Modularity for each algorithm. The Precision and Recall are not satisfactory criteria for community detection but the F-Measure is useful because it measures the level of similarity of the detected and the true communities. Infomap provided the best performance reflected on the highest value of the F-Measure ($F = 0.978$). This means that the detected communities are very similar to the true ones. However, the Modularity indicates relatively small differences between the two methods (Table 1) and thus this criterion can be ignored when choosing one of the algorithms.

Following the Step 2, Figure 16 shows the matrix of connections between communities detected using either Infomap or Louvain algorithms. The number of edges within the same community is considerably higher than the number of edges between communities,

Table 1 – Modularity, Precision, Recall and F-Measure for the Primary school network using Infomap and Louvain algorithms. For Precision, Recall and F-Measure, values vary from 0 to 1, and for Modularity, values vary from -1 to 1. A higher value indicates better performance. Reprinted from (LINHARES et al., 2020) ©2020 Springer Nature.

Criterion	Infomap	Louvain
Modularity	0.663	0.680
Precision	0.958	0.649
Recall	1.0	0.995
F-**Measure**	**0.978**	0.786

which indicates coherent results in 100% of the detected communities. Infomap was able to identify the true number of communities (10) in the network but the Louvain algorithm identified only 6 communities, i.e., some of the true communities were detected as a single community in this case.

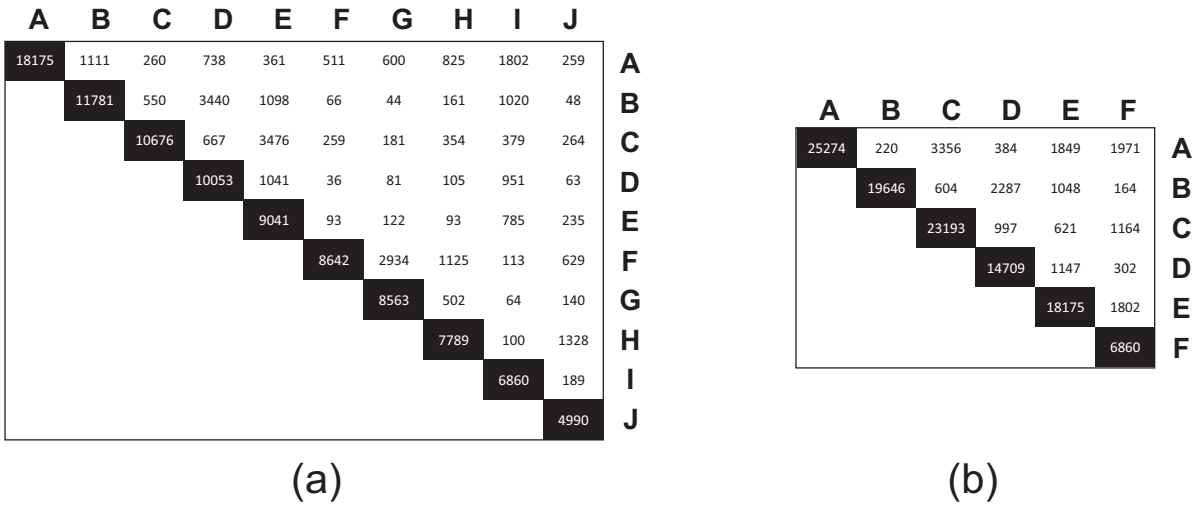

Figure 16 – Matrix of connections showing the number of edges between and within detected communities using (a) Infomap and (b) Louvain algorithms for the Primary school network. Reprinted from (LINHARES et al., 2020) ©2020 Springer Nature.

Analyzing the Step 3, we apply our proposed layout in the network data. Figure 17 shows detected communities using the different algorithms and the match with the true community of the node. Figure 17(a) shows two equivalent communities with 100% of common nodes, i.e., both algorithms detected communities with the same nodes. These communities correspond to classes 1A and 1B. Figures 17(b,c) show the non-equivalent communities from Louvain and Infomap, respectively. The connection matrix for Infomap showed that the numbers of identified and true communities are the same but did not show the community membership of each node (Fig. 16). However, our visualization methodol-

ogy provides this information and helps to quickly identify that each community correctly corresponds to a specific class and teacher (Fig. 17(a,c)). In addition, although the network did not explicitly inform which teacher belongs to a class, we are able to identify this association using Infomap and teachers correspond to black nodes in each community. This unique identification was not possible to find using the Louvain algorithm because almost every community has two teachers.

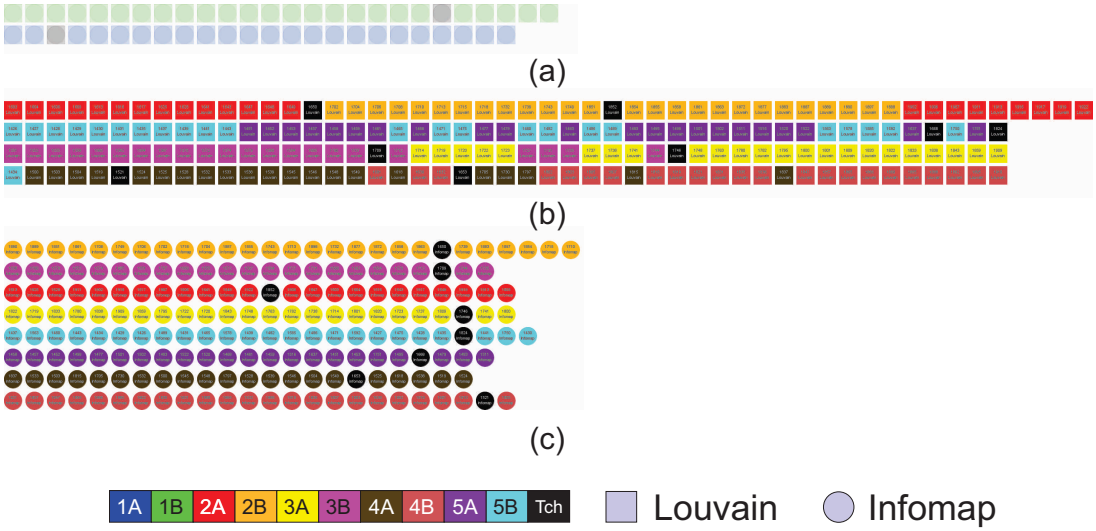

Figure 17 – Visualization of node membership in the detected and true communities of the Primary school network. Panel (a) shows the equivalent communities and the non-equivalent communities are shown on panels (b) for Louvain and (c) for Infomap algorithms. Circles and squares represent nodes and colors represent the true membership of each node, i.e., its classes. Black circles and squares correspond to teachers. Reprinted from (LINHARES et al., 2020) ©2020 Springer Nature.

The Louvain algorithm not only identified less communities than Infomap but also merged two classes of the same grade with the exception of the first grade (1A and 1B). As same-grade students interact more between themselves than with students from other grades (STEHLÉ et al., 2011), the result provided by the Louvain algorithm is also coherent. However, the direct links between class and teacher provided by Infomap suggest that this is the most appropriate algorithm to study this network.

Although the visual analysis supports that Infomap is more appropriate than Louvain for this network, it would be harder, without this visualization, to see that Louvain identified communities that are divided by grade whilst Infomap divided them according to classes together with the teacher and students in each one of them. Depending on the network analysis task, one algorithm may be more appropriate than the other. Such information may greatly influence decisions making. According to the analysis of our methodology steps, **Infomap** is more appropriate to be used in the Primary school network.

3.3.2 *High school* case

The second network corresponds to interactions between students in a high-school in Marseille, France (MASTRANDREA; FOURNET; BARRAT, 2015). The data were collected between the 2nd and 6th of December 2013 (i.e., five days). The network has 327 nodes and 188,508 edges and nine classes of students categorized in one out of four different types of specialization: MP, MP*1 and MP*2 focus on mathematics and physics; PC and PC* focus on physics and chemistry; PSI on engineering; and 2BIO1, 2BIO2, and 2BIO3 on biology. In this case, we also define the "true" community to specific classes because most interactions occur inside the classes (MASTRANDREA; FOURNET; BARRAT, 2015).

We evaluate the community detection algorithms using the methodology Step 1. Notice that the Louvain algorithm achieves better results for F-Measure (Table 2), suggesting that this algorithm is more appropriate to study this network according to this criterion. The modularity also indicates better results for the Louvain method, which also provides better detection of communities.

Table 2 – Modularity, Precision, Recall and F-Measure for the High-school network using Infomap and Louvain algorithms. For Precision, Recall and F-Measure, values vary from 0 to 1, and for Modularity, values vary from -1 to 1. A higher value indicates better performance. Reprinted from (LINHARES et al., 2020) ©2020 Springer Nature.

Criterion	Infomap	Louvain
Modularity	0.749	**0.822**
Precision	0.967	0.966
Recall	0.579	0.907
F-Measure	0.724	**0.936**

The analysis of Step 2 for the *High school* network shows coherent results, i.e., the higher values are present in the main diagonal in both detection methods. Then, this criterion does not help in the choice of the best detection method for this network. The number of communities detected by Louvain is 11 communities in this network (Fig. 18(a)). For comparison, Infomap returned 38 communities, as shown in Fig. 18(b).

For the Step 3, Figure 19 shows more clearly the difference in the number of communities detected by both methods and the distribution of nodes in each community. Infomap detected several communities with only two or three nodes (Fig. 19(c)) that is an undesirable characteristic which may demand further merging of these very small communities to make them meaningful (YIN et al., 2015). Moreover, these communities usually share the same label, suggesting that groups of students (possibly friends) interact frequently between them inside each class. On the other hand, the Louvain algorithm was more efficient than Infomap to identify the true communities linked to the nine classes

(Fig. 19(a,b)). The sizes of the equivalent communities and the communities identified by Louvain were similar as well, as seen by comparing Fig. 19(a) and Fig. 19(b).

The nodes highlighted in Fig. 20 represent the same nodes in both algorithms. These nodes are members of four large communities according to the Louvain detection (Fig. 20(a)) but are spread in six smaller communities when Infomap is used (Fig. 20(c)). This observation indicates that the results obtained by Louvain are more coherent since it grouped students in communities more closely related to their classes whereas Infomap detected more mixed communities (same-class nodes are more spread in various communities). According to the analysis of all steps, the **Louvain** algorithm is more appropriate to study the High-school network.

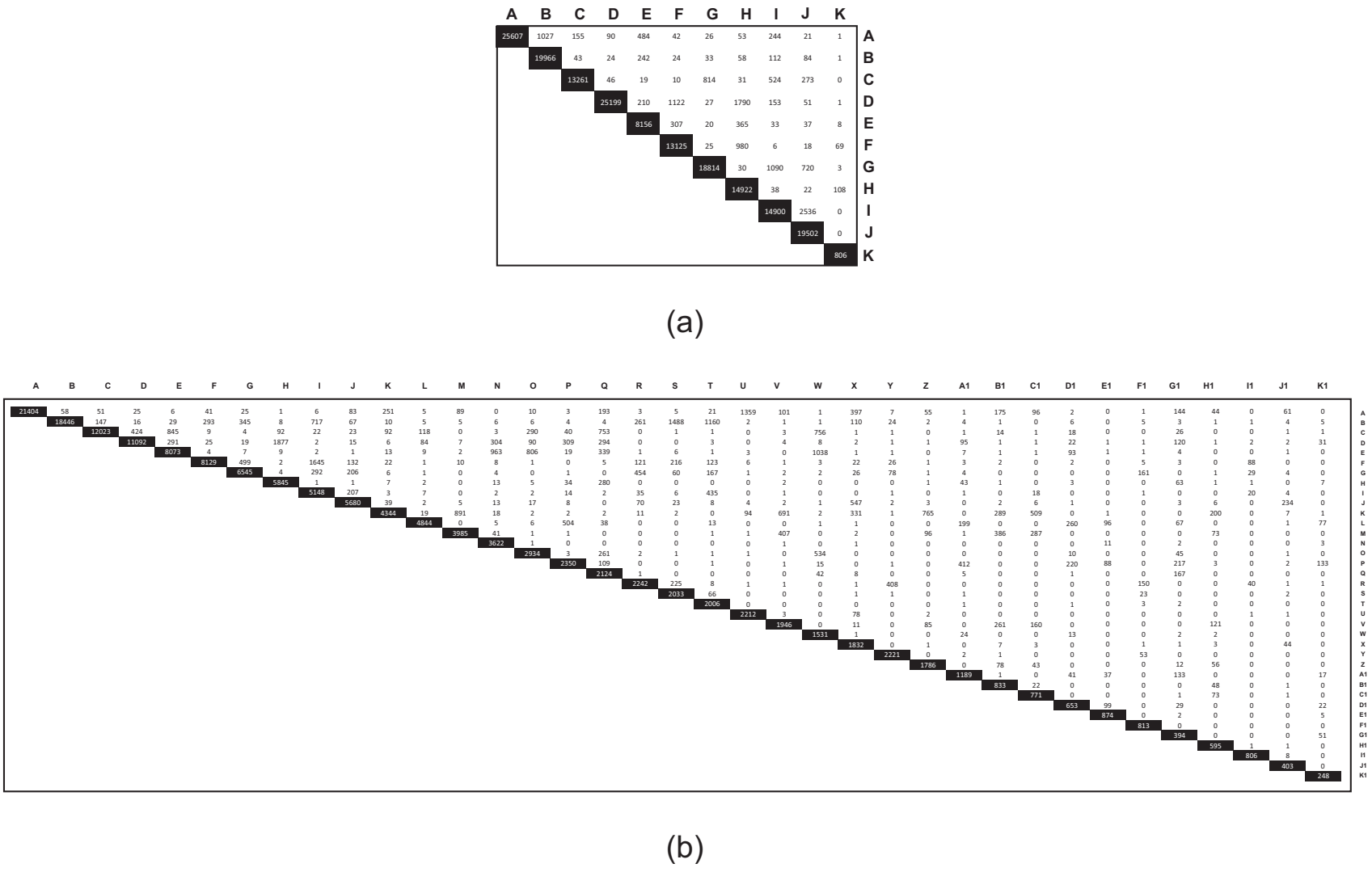

Figure 18 – Connection matrices for the *High School* network with the amount of interactions between the detected communities: (a) Louvain; (b) Infomap. Adapted from (LINHARES et al., 2020) ©2020 Springer Nature.

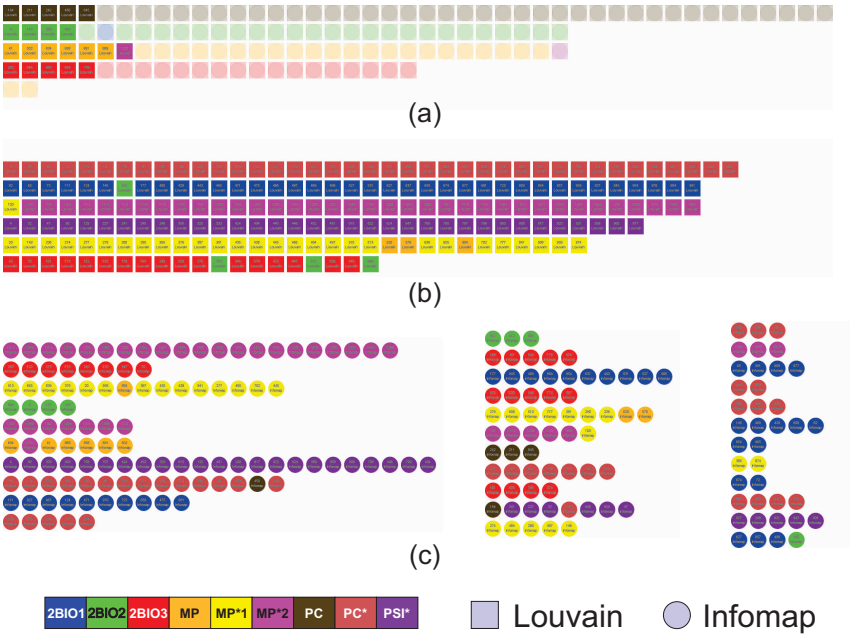

Figure 19 – Visualization of node membership in the detected and true communities of the High-school network. Panel (a) shows the equivalent communities and the non-equivalent communities are shown on panels (b) for Louvain and (c) for Infomap algorithms. Circles and squares represent nodes and colors represent the true membership of each node, i.e., its classes. Reprinted from (LINHARES et al., 2020) ©2020 Springer Nature.

3.3.3 *Hospital* case

The third network corresponds to interactions between patients and staff in the geriatric unit of a university hospital in Lyon, France (VANHEMS et al., 2013). The data collection occurred between the 6th and 10th of December 2010 (i.e., five days). The network contains 75 nodes and 32,424 edges. Nodes represent 29 patients and 46 hospital staff classified in different profiles: 11 people are medical doctors (physicians or interns - MED), 27 are nurses or nurses' aides (NUR) and 8 are administrative staff (ADM). For complementarity, all patients were associated with the PAT profile.

In this network, there is no direct relationship between the communities and the people's profiles since a high level of interactions between individuals from different profiles is expected. Therefore, we are unable to define "true" communities with the available information and thus we can not use the F-Measure in the methodology Step 1 to determinate which algorithm has the best performance in this network. It is, however, possible to analyze the modularity of the network since the modularity does not depend on the label of nodes. Table 3 indicates that the Louvain algorithm shows superior performance in comparison to Infomap with a better distribution of nodes in communities.

Figure 21(a) shows the matrix of connections (Step 2) between communities when Infomap is used. This gives inconsistent results for five out of seven detected communities,

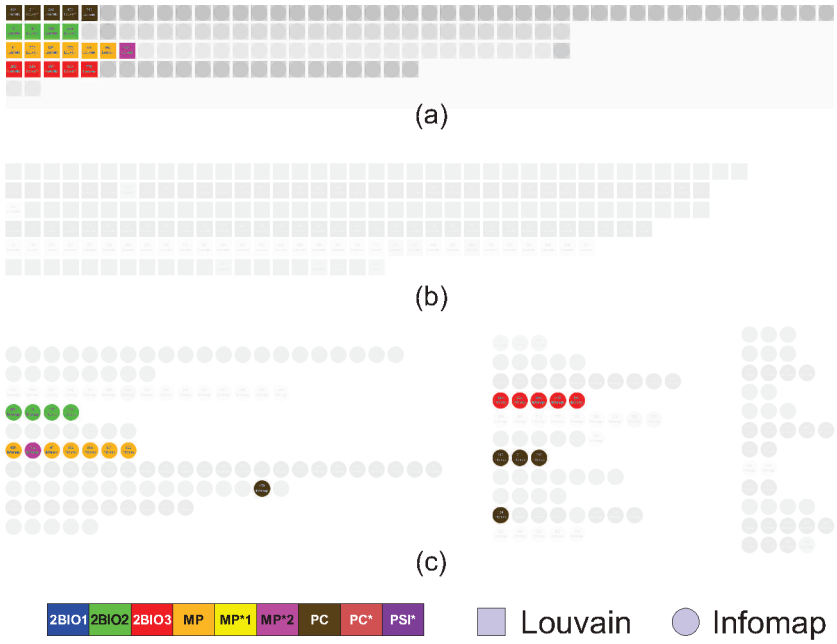

Figure 20 – Visualization of selected nodes in detected communities of the High-school network. Panel (a) shows equivalent communities and the non-equivalent communities are shown on panels (b) for Louvain and (c) for Infomap algorithms. The colors represent the classes, i.e., true communities. Reprinted from (LINHARES et al., 2020) ©2020 Springer Nature.

Table 3 – Modularity for the Hospital network using Infomap and Louvain algorithms. Modularity values vary from -1 to 1, and higher value indicates better performance. Reprinted from (LINHARES et al., 2020) ©2020 Springer Nature.

Criterion	Infomap	Louvain
Modularity	0.313	**0.397**

representing 71.43% of the communities. The nodes from communities C, D, E, F, and G have more edges with nodes from community A than with nodes from their communities. The analysis of the Louvain matrix also indicates inconsistent result for three out of the six identified communities, i.e., in 50% of the communities (Fig. 21(b)), suggesting that the Louvain algorithm produces better results than Infomap.

According to the Step 3, applying Infomap on this network results in a number of communities with one relatively very large and several very small ones (with only two or three nodes) (Fig. 22(c)). This distribution of nodes in communities may explain why five out of seven communities identified by Infomap have more connections with other communities than within themselves (Fig. 21(a)). As mentioned before, the detection of very small communities is not good because communities with a few nodes are not meaningful. A visual analysis of the distribution of nodes generated by both algorithms shows that the largest community identified by Infomap is divided into smaller ones by Louvain (Fig. 22(b) and Fig. 22(c)). Although we expected a priori a more homogeneous

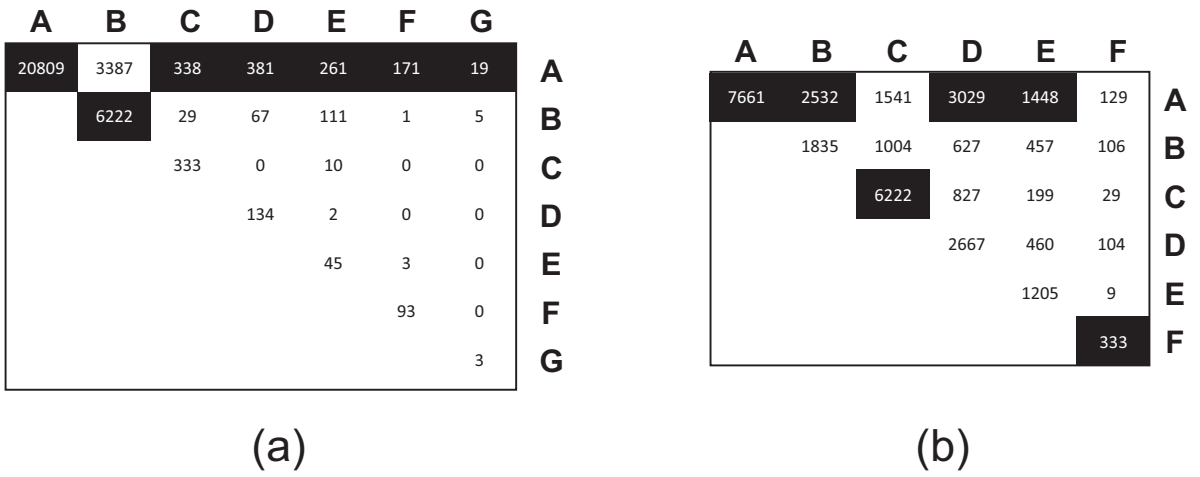

Figure 21 – Matrix of connections showing the number of edges between and within detected communities using (a) Infomap and (b) Louvain algorithms for the Hospital network. Reprinted from (LINHARES et al., 2020) ©2020 Springer Nature.

distribution of profiles in the detected communities, this is not the case for the MED profile in any of the algorithms, as seen in the equivalent communities (Fig. 22(a)). Both algorithms identified the very same community for medical doctors. This pattern may reflect the fact that the network represents a university hospital where medical doctors frequently interact with interns resulting in grouping them in the same community. Louvain also shows more coherent results (Fig. 22(b)) as the community sizes are more similar and communities better reflect the interactions between nurses and patients (VANHEMS et al., 2013). Our analysis of all steps supports that the **Louvain** algorithm is more appropriate for the *Hospital* network.

3.3.4 *InVS* case

The fourth network corresponds to interactions between the staff of the French Institute of surveillance in public health (GÉNOIS et al., 2015). This institute is organized in five departments: Scientific Direction (DISQ), Chronic Diseases and Traumatisms (DMCT), Health and Environment (DSE), Human Resources (SRH) and Logistics (SFLE). The data were collected during two weeks (ten business days) in 2013. The network has 92 nodes and 9,827 edges. There are two types of nodes: "residents" that are nodes mostly interacting with nodes from their departments, and "travelers" that are nodes mostly interacting with nodes from other departments. Most nodes are "residents".

In this network, with the exception of the "travelers" nodes, communities are strongly related to departments (GÉNOIS et al., 2015). Considering this relation between communities and departments, we define each department as the "true" community for the eval-

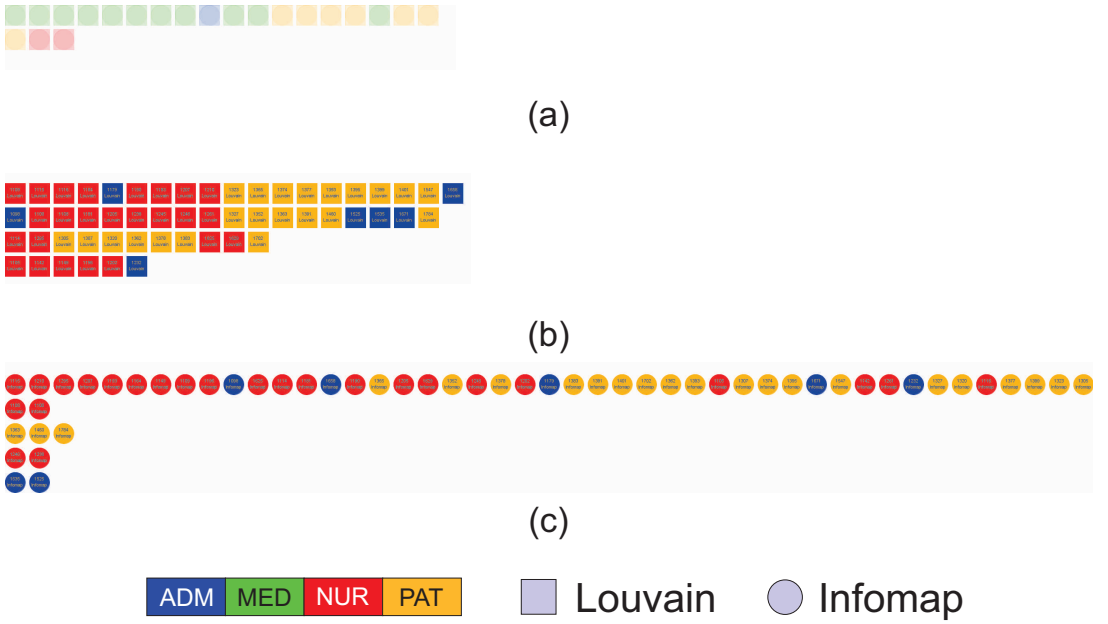

Figure 22 – Visualization of node membership in the detected communities of the Hospital network. Panel (a) shows the equivalent communities and the non-equivalent communities are shown on panels (b) for Louvain and (c) for Infomap algorithms. Circles and squares represent nodes and colors represent the profile of each node. Reprinted from (LINHARES et al., 2020) ©2020 Springer Nature.

uation of the community detection algorithms using methodology Step 1. Table 4 shows that Louvain has the best performance with the highest value for F-Measure ($F = 0.883$). As for the Primary school, modularity gives slightly better results with Louvain than with Infomap. Since it does not indicate significant differences between both algorithms, this criterion may be ignored when choosing one of the algorithms.

Table 4 – Modularity, Precision, Recall and F-Measure for the InVS network using Infomap and Louvain algorithms. For Precision, Recall and F-Measure, values vary from 0 to 1, and for Modularity, values vary from -1 to 1. A higher value indicates better performance. Reprinted from (LINHARES et al., 2020) ©2020 Springer Nature.

Criterion	Infomap	Louvain
Modularity	0.631	0.645
Precision	0.954	0.897
Recall	0.774	0.869
***F*-Measure**	0.855	**0.883**

In the Step 2, the matrix of connections for this network shows that both algorithms have good results since most of the detected communities have more internal connections than connections with other communities (Fig. 23). The only exception occurs with the matrix M of Infomap, where $M[J, J] < M[C, J]$ indicates that the nodes of community

J have more connections with nodes from community C than with nodes from their own community J (Fig. 23(a)). As the matrix for the Louvain algorithm shows coherence for all communities (main diagonal of Fig. 23(b)), it gives the most consistent results for InVS network according to this criterion.

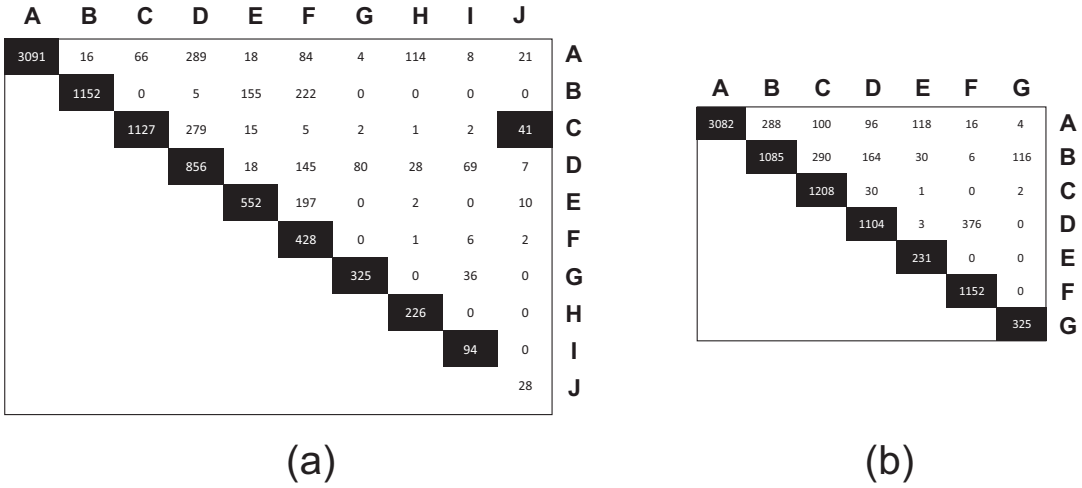

Figure 23 – Matrix of connections showing the number of edges between and within detected communities using (a) Infomap and (b) Louvain algorithms for the InVS network. Reprinted from (LINHARES et al., 2020) ©2020 Springer Nature.

Figure 24 shows the distribution of nodes in communities (Step 3). Although most communities detected by both algorithms are equivalent (Fig. 24(a)), Infomap (Fig. 24(c)) detected more communities than Louvain (Fig. 24(b)) but several of the communities have a few nodes only. Some nodes have different colors than other nodes in the same community, as for example, the orange node in the yellow community in Fig. 24(b). This pattern may be either noise or represent "traveler" nodes. Since both algorithms generated similar results considering the number of equivalent communities, it is unclear which is the best algorithm to support the visual analysis of the InVS network. Even in this case, the proposed layout highlights the differences between the algorithms and helps users to choose the preferred solution: if the user prefers to perform fine-granularity analysis, i.e., with more communities, the best choice is Infomap; complementary, the user can also choose Louvain as it presents slight better results in F-Measure. Therefore, since both algorithms show coherent results, there is no best algorithm to study this network according to the methodology steps.

3.4 Discussion

Table 5 summarizes basic information about the networks used in the case studies. Each network has different properties and structures which contribute to evaluating the community detection algorithms in different contexts.

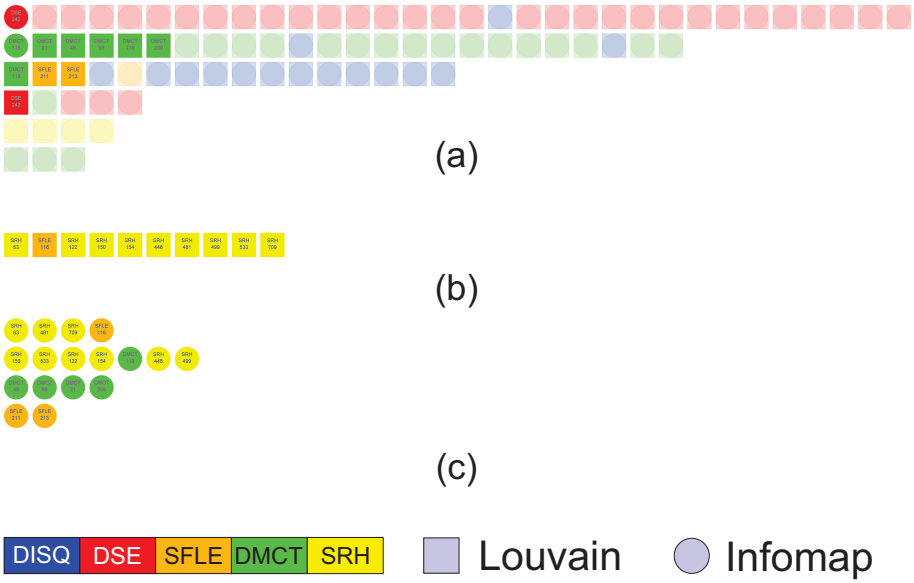

Figure 24 – Visualization of node membership in the detected and true communities of the InVS network. Panel (a) shows the equivalent communities and the non-equivalent communities are shown on panels (b) for Louvain and (c) for Infomap algorithms. Circles and squares represent nodes and colors represent the true membership of each node, i.e., its department. Reprinted from (LINHARES et al., 2020) ©2020 Springer Nature.

Table 5 – Fundamental properties of each studied network (Com.: Communities). Reprinted from (LINHARES et al., 2020) ©2020 Springer Nature.

Network	#Nodes	#Edges	#Labels	#Com. Infomap	#Com. Louvain
Primary school	242	125.773	11	10	6
High-school	327	188.508	9	38	11
Hospital	75	32.424	4	7	6
InVS	92	9.827	5	10	7

To illustrate the diversity of the four analyzed networks, Figure 25 shows the node-link diagram for each network. The node-link diagram reveals several characteristics of the networks and helps to identify two types of behaviors: (i) networks in which the nodes are well visually separated in relation to their labels (e.g., departments or classes – Fig. 25(b,d)) and (ii) networks in which the organization of nodes in the layout does not easily allow the identification of this separation between detected and true communities (Fig. 25(a,c)). This last observation is expected in networks with high mixing between nodes, as for example in the Hospital (Fig. 25(c)), where the number of interactions between different profiles is sometimes larger than the interactions within the same profile.

The network visual analysis with both community detection algorithms allows identifying common patterns across all four networks. First of all, Infomap identified more communities than Louvain in all cases (see Table 5). Moreover, the probability of having communities with just a few nodes increases when the number of identified communities

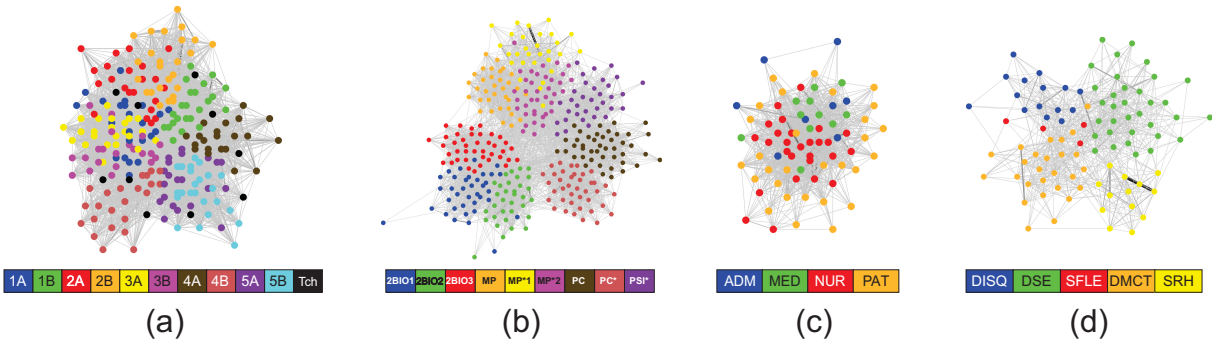

Figure 25 – Node-link diagram layout of each network: (a) Primary school; (b) High-school; (c) Hospital; (d) InVS. The colors represent the node labels. Reprinted from (LINHARES et al., 2020) ©2020 Springer Nature.

is larger, as occurs in all four networks but the Primary school with Infomap.

Another common pattern is related to the quality of the detection algorithms: Louvain was superior in two out of the four networks (Hospital and High-school) according to the tested criteria. The connection matrices between and within communities showed coherent results for both algorithms in three out of the four networks, except in the Hospital network. This result reflects the good quality of the response of the selected algorithms as already reported in the literature using different analysis (FORTUNATO; HRIC, 2016). Finally, the matrix of connections also helped to analyze the Hospital network, where no "true" community structure was available for testing.

Figure 26 shows that the number of equivalent communities increases as T_{Eq} decreases for all studied networks. Figure 26 also shows that the number of equivalent communities is equal to the total number of communities identified by Louvain (the algorithm that returned fewer communities – see Table 5) if $T_{Eq} = 0\%$ for all networks but Hospital. The inconsistent results of the community detection for the Hospital network (Section 3.3.3) may explain why this network was particular.

Figure 27 shows the dependence of the layouts with T_{Eq} for the High-school network, that was the network with the highest variation in the number of equivalent communities (Fig. 26). With $T_{Eq} = 0\%$, all communities from Louvain are considered as equivalent (Fig. 27(a)). The number of equivalent communities decreases with higher T_{Eq} until the minimum value with $T_{Eq} = 100\%$. This result is expected, as discussed in Section 3.2.1.

Community detection algorithms with computational linear complexity, such as Infomap and Louvain, are often used in large networks with thousands of nodes and edges. The visualization method and layout proposed in this thesis does not address scalability, which is a challenging limitation and requires further studies. In fact, visual scalability of networks remains a major and open problem in the area of Information Visualization (BURCH, 2017).

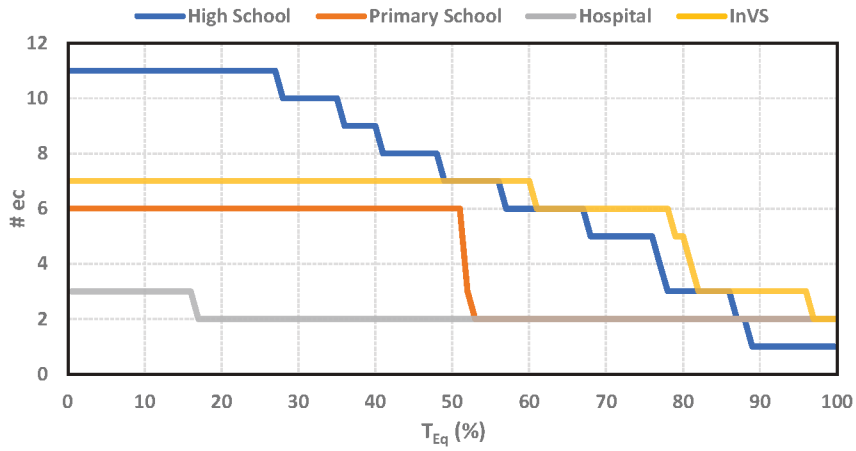

Figure 26 – Variation of the number of equivalent communities ($\#ec$) and T_{Eq} for all four networks. Reprinted from (LINHARES et al., 2020) ©2020 Springer Nature.

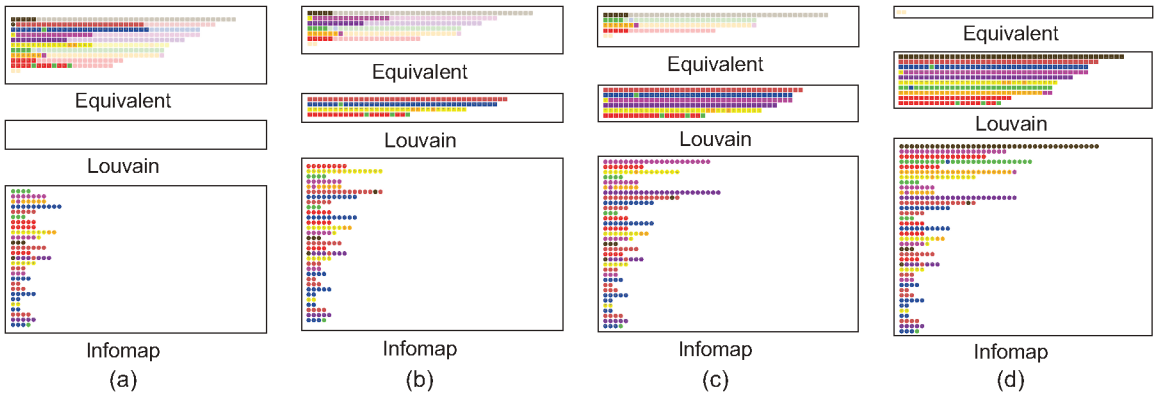

Figure 27 – Visualization of the distribution of nodes in communities for the High-school network for (a) $T_{Eq} = 0\%$, (b) $T_{Eq} = 50\%$, (c) $T_{Eq} = 75\%$, (d) $T_{Eq} = 100\%$. Reprinted from (LINHARES et al., 2020) ©2020 Springer Nature.

3.5 Final considerations

Network community detection algorithms are well established in the literature and are used in several contexts to study real-world network data. However, the choice of the best algorithm for a specific task may not be straightforward. This thesis presented a methodology to evaluate two generic community detection algorithms by using statistical and visual analysis to support decision making. We tested the performance of our methodology with four case studies by using the popular Louvain and Infomap algorithms. Our analysis identified similarities and differences between the algorithms using both statistical measures and visual analysis.

Our study demonstrates the importance and usefulness of visual network analysis. Statistical analysis is sometimes a "black-box" and challenging for some users, particularly qualitative-oriented researchers and practitioners, and decisions on the best algorithms based on numerical results may be difficult. The central role of the user in the decision

making process employing Information Visualization techniques is important. It facilitates deciding on the best community detection algorithm for the specific network under analysis, enhancing the understanding of the network and the identification of relevant patterns.

According to the evaluation involving the three main steps of methodology, none of the tested community detection algorithms was the best in all four case studies. However, Louvain was superior in two out of the four case studies and Infomap in only one of them. We concluded that both algorithms perform similarly for the InVS network. Even in this case, our visualization methodology was able to provide valuable insights that allow users to choose which algorithm is more suitable for their purposes.

In the future, we plan to improve the study of community visual analysis, especially focusing on experiments involving domain-specific users and tasks. The goal is to identify whether the best algorithm, evaluated through statistical analysis, shows the best user visual experience. We plan to extend the analysis and include other well-established community detection algorithms and larger networks with more nodes and edges. Also, we intend to test other statistical measures, such as Triangle Participation Ration (TPR) and Adjusted Rand index. For the equivalent communities detection, we intend to use popular matching strategies, such as Normalized Mutual Information (NMI) and variational information.

Chapter 4

A Scalable Strategy for Temporal Network Visualization

This chapter describes the Community-based Node Ordering (CNO), a novel scalable method to enhance the MSV layout. The method employs the community notion in order to approximate related nodes and to reduce the visual clutter, to validate our hypothesis that state that visualization techniques that receive the network community structure as input reduce visual clutter and thus enhance temporal network analysis. Additionally, a taxonomy is proposed to categorize communities according to their activity patterns. Furthermore, a quantitative analysis is defined to evaluate the proposed methods in relation to others. This chapter is based on the published article (LINHARES et al., 2019b). The final authenticated version is available online at: <http://dx.doi.org/10.1016/j.cag.2019.08.006>.

4.1 Visual scalability

Due to the large amount of information on temporal networks, statistics and topological measures are typically used to characterize the dynamic structure of the network to identify relevant patterns. However, the use of information visualization strategies help researchers to identify trends and anomalies faster and thus to better understand the interactions between the nodes. In MSV layout, the positions of nodes affect the visualization and thus the identification of structural and activity patterns. The main limitation of this layout is the potential edge overlap at each time step. Previous research has shown that node reordering techniques can reduce this overlap and thus visual clutter, highlighting or revealing hidden patterns (ELZEN et al., 2014; LINHARES et al., 2017b). However, previous techniques are not visually scalable and limited to relatively smaller networks (ELZEN et al., 2014; MI et al., 2016).

4.2 Community-based Node Ordering

The *Community-based Node Ordering (CNO)* strategy was designed to enhance the identification of visual patterns in temporal layouts. It is based on the assumption that nodes with more edges between themselves should be spatially close to reduce the edge sizes and consequently their overlap and visual clutter. The three steps of the CNO algorithm are the following:

Step 1 - Community detection: To detect and categorize nodes in non-overlapping communities using any community detection algorithm in the aggregated static network.

Step 2 - Inter-community reordering: To order the communities in the layout, employing any node reordering strategy by assuming a community as a single node. This step reduces the edge sizes between communities and highlights edge patterns.

Step 3 - Intra-community reordering: To order the nodes in each community, employing any node reordering strategy. This step further reduces edge overlap by approximating pairs of nodes sharing many edges inside the respective community.

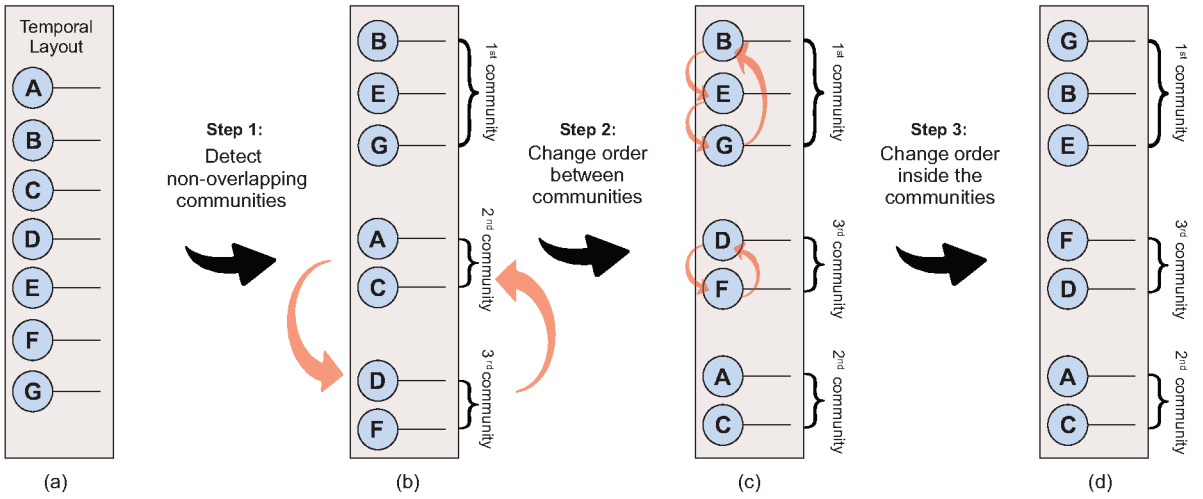

Figure 28 – Example of the *Community-based Node Ordering (CNO)* strategy considering the first level. CNO combines an algorithm to detect communities with node and community reordering algorithms. (a) Original network; (b) Non-overlapping communities detected in Step 1; (c) Sequence of communities after the inter-community reordering step (Step 2); (d) Final sequence of nodes returned by the intra-community reordering step (Step 3). Reprinted from (LINHARES et al., 2019b) ©2019 Elsevier.

In networks with thousands of nodes and edges, the CNO can be recursively applied n times (i.e., up to n levels) or until no more communities can be decomposed. This hierarchical algorithm controls the level of granularity (or resolution) to analyze the data in

the layout. For example, communities could correspond to cities in the first level, neighborhoods in the second, then streets, and households (FORTUNATO; BARTHLEMY, 2007). Algorithm 1 presents the pseudo-code for the recursive CNO. Moreover, the visual analysis may be improved through user interactions in the resulting CNO layout.

Algoritmo 1 CNO

Input: Network, CurrentLevel

```
C ← CommunityDetection(Network);
if |C| > 1 then
    InterCommunityReordering(C);
    foreach community c_i ∈ C do
        CNO(c_i, CurrentLevel + 1);
    end
else
    IntraCommunityReordering (C);
end
```

CNO allows the interaction with the layout at different levels (Figure 29(b-d)). To focus the analysis, the user may also clean the layout by filtering specific edges, which can be done in two ways. The first one is to show only the inter-community edges, i.e., only those edges connecting nodes that belong to different communities (Figure 29(e)). This strategy helps to analyze the interactions between communities over time. The second one aims to analyze the interactions among nodes within the same group, by showing only edges that connect nodes in same community (Figure 29(f)). Despite the loss of information caused by the edge filtering, patterns that were not visible due to the visual clutter may now be revealed. These particular edge filtering algorithms cannot be applied in other node reordering strategies if the nodes are not organized into communities or groups.

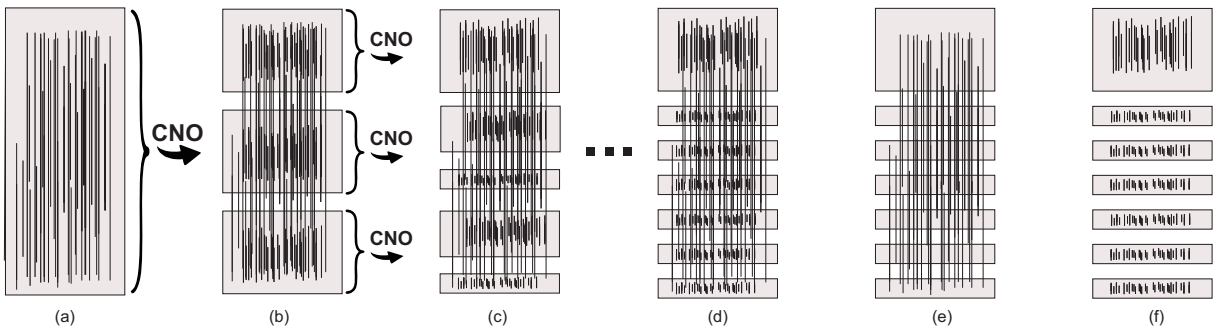

Figure 29 – Interacting with the CNO layout: (a) Network with the original node ordering; (b) CNO level 1; (c) CNO level 2; (d) CNO level n; (e) CNO level n showing only inter-community edges; (f) CNO level n showing only intra-community edges. Reprinted from (LINHARES et al., 2019b) ©2019 Elsevier.

The CNO running time is defined by the time complexity of the chosen community detection algorithm and the node reordering strategy. Both community detection algorithms used in our experiments, as it will be explained further, (Infomap and Louvain) are $\mathcal{O}(m)$, where m is the number of edges in the network (FORTUNATO, 2010). For node reordering, the best strategy to reduce visual clutter is the RN, which is $\mathcal{O}(n * m)$, where n is the number of nodes. Therefore, the CNO complexity is mainly dependent on the node reordering and the bottleneck is the number of edges. The CNO is flexible and accommodates any non-overlapping community detection algorithm and node reordering strategy. Depending on the task and network density, faster methods, such as Lexicographic, Degree, or even random placement of nodes (that has linear complexity) could be used; the drawback however is the potential reduction of layout quality due to higher edge overlap.

4.3 Layout analysis and evaluation

In order to improve the temporal analysis, we propose a taxonomy to visually categorize different types of communities. Also, three measures are described to support a quantitative analysis of the layouts, providing a tool for comparison between CNO and other proposals.

4.3.1 Taxonomy for communities categorization

Taxonomies have been proposed to categorize community detection methods (DRIF; BOUKERRAM, 2014), networks according to features of their communities (ONNELA et al., 2012), and the principles that underpin the community detection (ROSVALL et al., 2019). Our taxonomy categorizes different types of communities in the CNO layout according to their activity patterns. The user is then able to choose communities and focus the analysis, avoiding distractions and accelerating decision making. The three categories that form this taxonomy are based on activity frequency, dispersion, and type:

- **Activity Frequency - Sporadic or Continuous**: There may be edges (i.e., activity) between nodes within a community during the entire time interval or only in specific periods of time. To distinguish these scenarios, the time interval of the network ($T_{\text{Final}} - T_{\text{Initial}}$) is divided into sub-intervals of length S_I. If at least one of these intervals has no edges, the community has sporadic activity (Figure 30(a)), which can be an *Initial* one (when low or no activity is observed close to T_{Initial}), a *Final* one, when low or no activity is observed close to T_{Final}, or neither of them (low or no activity is observed in the middle of the network). If all intervals have at least one edge each, the community is classified as having continuous activity

(Figure 30(b)). This classification is useful, for example, to identify the activity of edges that are interrupted at times, such as seasonal events.

- **Activity Dispersion - Dispersed or Grouped**: A community has grouped activity if the majority of consecutive edges is too close to each other (according to a threshold T_{dg}), as depicted in Figure 30(d), or has dispersed activity otherwise (Figure 30(c)). For example, imagine a network in which the nodes are clients and pizzerias, and the edges represent sales of pizzas. Assume that pizzas are sold every day and so there is continuous activity in this network. However, continuous activity does not imply in a successful business and thus it is important to consider the number of pizzas sold each day to know if the pizzeria is successful (selling several pizzas per day, which represents a large number of edges close to each other, i.e., grouped activity) or if it should be closed (it is selling few pizzas per day and thus has few and sparse edges over time, i.e., dispersed activity).

- **Activity Type - Homogeneous or Heterogeneous**: The previous categories are generic and can be applied to any temporal network but the classification of a community in homogeneous or heterogeneous depends on additional categorical information about the edges. Using edge metadata, we perform the following classification: if the majority of the edges in a community (more than a threshold T_h) has the same category, the community is assumed homogeneous (Figure 30(e)); otherwise, it is heterogeneous (Figure 30(f)). Considering an email network as example, a community would have homogeneous activity if the majority of its edges represent a specific type of message (e.g., personal or professional).

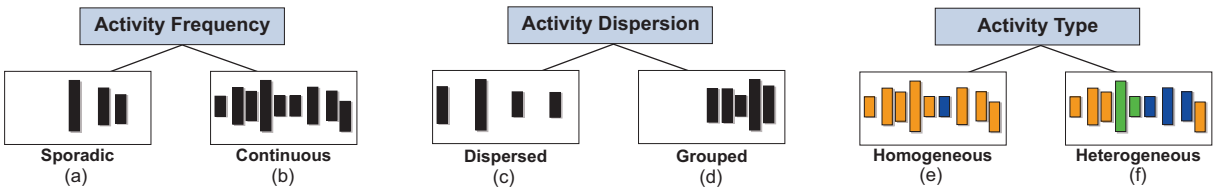

Figure 30 – Taxonomy proposed to classify each community in the MSV layout according to its activity patterns. Reprinted from (LINHARES et al., 2019b) ©2019 Elsevier.

These categories provide a more detailed view of the network and improve the visual analysis, allowing the user to focus on communities that fit in specific categories of the taxonomy. Figure 31 presents four examples of layouts generated by combining the "Activity Frequency" and "Activity Dispersion" categories. A layout activity is defined as Sporadic and Grouped when all sets of edges are dense and not continuous over time (i.e., seasonal edges), as seen in Figure 31(a). Similarly, a layout has Sporadic and Dispersed

activity when there is at least one time interval without edge and no dense areas in the layout (Figure 31(b)). A continuous layout is either grouped (when the majority of consecutive edges is close to each other), as illustrated in Figure 31(c), or dispersed (when such edges are not so close to each other), as shown in Figure 31(d). Depending on the network and chosen thresholds, a few communities are classified as having Continuous and Dispersed activity due to the particularities of this scenario.

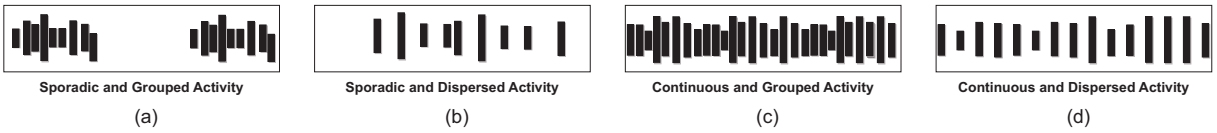

Figure 31 – Examples of different layouts according to the "Activity Frequency" and "Activity Dispersion" categorizations. Reprinted from (LINHARES et al., 2019b) ©2019 Elsevier.

The chosen temporal resolution affects the output of the taxonomy classification. For example, a network may be classified as dispersed activity when using one hour resolution, or as grouped activity when a day resolution is chosen. Such variations provide a more holistic understanding of the temporal patterns.

4.3.2 Quantitative analysis

Each node reordering method may result in a different layout as its sequence of nodes can be different from each other (Fig. 32). In order to evaluate the quality of a layout in terms of visual clutter and compare different reordering algorithms, we have considered three measures suitable for the MSV layout: number of overlapping edges (*# Overlapping Edges*); average edge size (*Avg Edge Size*); and number of intersections (*# Intersections*).

The number of overlapping edges is an important feature but it is not possible to measure the visual clutter of a temporal layout by knowing only its value since it does not take into account the edge sizes. We define an edge size as $n + 1$, where n is the number of nodes positioned between its two nodes, without counting them. Similarly, the average edge size is not enough to evaluate a layout as it does not indicate whether a region is dense (i.e., visually cluttered). The number of intersections is a measure that counts the number of parts of the edge overlapping other edges. Two edges that have several intersections generate more visual clutter than two edges with fewer intersections. A higher number of intersections is expected in larger networks. To simplify the representation and facilitate the analysis of different layouts, one assumes the layout with more intersections as a baseline and analyzes the performance (reduction on visual clutter) against others.

Figure 33 illustrates the application of these three measures. The three edges from timestamp 2 are positioned side-by-side to facilitate the understanding. The number of overlapping edges in this example is three, i.e., there are no overlapping edges at times

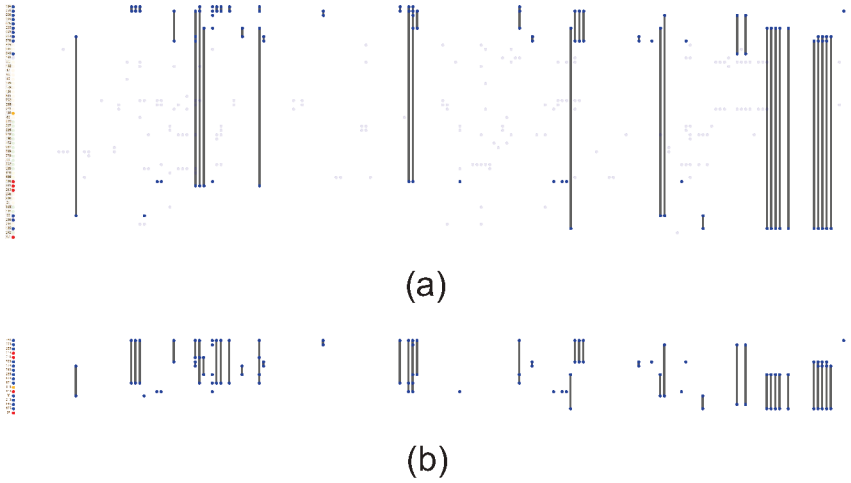

Figure 32 – Two layouts generated by different node reordering algorithms, *Appearance* in (a) and *Recurrent Neighbors* in (b), for the same subset of nodes. The edges sizes vary depending on the nodes position.

0 and 1, but there are three at time 2. The size of the edge between A and B at time 0 (A,B,0) is 1, the size of (C,E,0) is 2 and so on. The average edge size of the network is thus 2.28. Finally, the number of intersections in this network is 5 because there are five regions of intersection involving the three overlapping edges at time 2.

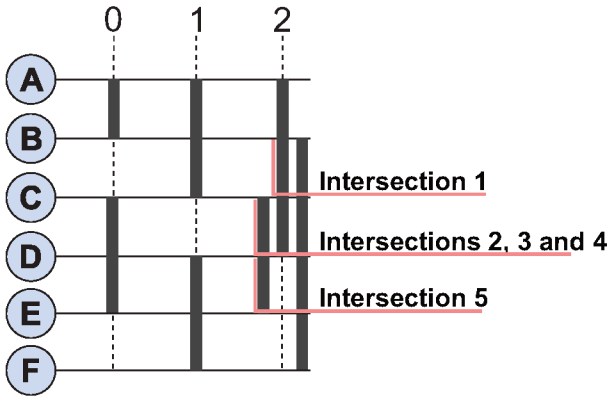

Figure 33 – Example of application of the analysis in a temporal layout. The three edges from timestamp 2 are positioned side-by-side to facilitate the understanding. There are 3 overlapping edges, the average edge size is 2.28, and there are 5 intersections. Reprinted from (LINHARES et al., 2019b) ©2019 Elsevier.

4.4 Case studies

We evaluated CNO using Infomap and Louvain (Step 1), which are two community detection algorithms highly recommended in the literature (LANCICHINETTI; FORTUNATO, 2009). In Steps 2 and 3, two efficient node reordering methods (*Degree* and *Recurrent Neighbors*) (LINHARES et al., 2017b) were used. Quantitative and visual anal-

ysis were used to test the performance of our method and to validate the quality of the CNO layout in relation to four node reordering strategies (*Appearance*, *Lexicographic*, *Degree* and *Recurrent Neighbors*). Experiments were performed using a software developed by the authors (Sec. 7), which implements CNO and all interactive tools and measures described in this thesis. Two real-world networks were analyzed to evaluate the visual scalability of CNO and its improvement over other strategies. The first data set, called Hospital network, is a relatively small network and was chosen to test a single CNO level (level 1). The community structure in this network is non-trivial. The second data set, called Twitter network, is a large data set and was chosen to illustrate the use of multiple CNO levels and the categorization of communities using the proposed taxonomy.

4.4.1 Small dataset – Hospital

The Hospital network (VANHEMS et al., 2013) records the contact (by proximity) between people on a hospital in Lyon, France (this is the same network presented in Sec. 3.3.3). The data refer to a geriatric unit and include the participation of 75 individuals, being 29 patients (PAT), 11 physicians (MED), 27 nurses and nurses' aides (NUR), and 8 administrative staff (ADM). The data set was collected using radio-frequency badges between December 6, 2010 (Monday) and December 10, 2010 (Friday) and has a total of 75 nodes and 11,977 edges distributed in the 5 days. The original sampling resolution is 20 seconds and we changed it for convenience. Table 6 shows a quantitative analysis of various node reordering algorithms applied to the Hospital network using measures from Section 4.3.2. Given the high density of edges (11,977 edges in total), overlapping is also relatively high with over 11,200 edges overlapping.

Table 6 – Quantitative analysis using different node reordering algorithms for the Hospital network (75 nodes and 11,977 edges). Each CNO configuration is represented by CNO (S_1, S_2, S_3), where S_x is the method used in Step x. The analysis was performed using level 1. RN: Recurrent Neighbors. Reprinted from (LINHARES et al., 2019b) ©2019 Elsevier.

Node Reordering Algorithm	# Overlapping Edges	Avg Edge Size	# Intersections	Less visual clutter than Appearance (%)
Appearance	11,573	22.92	499,841	—
Lexicographic	11,616	20.65	405,997	18.77
Degree	11,577	16.01	284,096	43.16
RN	11,337	12.99	209,052	58.18
CNO (Infomap, RN, RN)	11,262	12.33	198,907	60.21
CNO (Louvain, RN, RN)	11,206	14.57	226,385	54.71
CNO (Infomap, Degree, RN)	11,263	12.11	194,765	61.03
CNO (Louvain, Degree, RN)	11,346	17.25	297,730	40.44
CNO (Infomap, RN, Degree)	11,332	15.15	254,278	49.13
CNO (Louvain, RN, Degree)	11,275	15.79	246,119	50.76
CNO (Infomap, Degree, Degree)	11,332	14.97	250,662	49.85
CNO (Louvain, Degree, Degree)	11,411	18.28	314,455	37.09

Table 7 – Quantitative analysis using Appearance as a baseline and the intra- and inter-community filtering for the Hospital network (75 nodes and 11,977 edges). The analysis was performed using level 1. RN: Recurrent Neighbors; Intra: Intra-community edge filtering; Inter: Inter-community edge filtering. Reprinted from (LINHARES et al., 2019b) ©2019 Elsevier.

Node Reordering Algorithm	# Overlapping Edges	Avg Edge Size	# Intersections	Less visual clutter than Appearance (%)
Appearance	11,573	22.92	499,841	—
CNO (Infomap, RN, RN) Intra	8,777	7.91	78,519	84.29
CNO (Infomap, RN, RN) Inter	1,776	33.79	59,036	88.19
CNO (Louvain, RN, RN) Intra	3,917	2.8	7,754	98.45
CNO (Louvain, RN, RN) Inter	5,519	26.7	179,083	64.17

The number of overlapping edges does not take into account the edge sizes, so methods with a similar number of overlapping edges may result in different layouts due to the sizes of such edges. Table 6 shows that the *Recurrent Neighbors* algorithm generates shorter edges (12.99 on average) than *Appearance* (22.92), *Lexicographic* (20.65) and *Degree* (16.01). Since the *Appearance* method generates the layout with more visual clutter (with more intersections than the others – Table 6), we take it as a baseline to quantify the improvement given by the other methods. The naive strategies have the worst results in comparison to most of the other algorithms. While *Lexicographic* generates a layout with 18.77% less intersections than *Appearance*, the *Recurrent Neighbors* method is significantly cleaner (58.18% less intersections).

The quality of the results depends on the methods used in each step of the CNO algorithm. Table 6 shows that *CNO (Infomap, RN, RN)* and *CNO (Infomap, Degree, RN)* are more efficient than *Recurrent Neighbors* according to all adopted measures. On the other hand, *CNO (Louvain, Degree, Degree)* is only 37.09% cleaner than *Appearance*, a result worse than the ones from RN and *Degree*. By fixing the reordering methods used in Steps 2 and 3 and varying the community detection algorithm, as depicted, the average number of intersections of *CNO (Infomap, RN, RN)* and *CNO (Louvain, RN, RN)* are smaller in comparison to other node reordering combinations.

Table 7 shows the results using *Appearance* (as baseline) and CNO. The analysis considers both intra- and inter-community edge filtering, varying the community detection algorithm and adopting *Recurrent Neighbors* as node reordering strategy. Filtering the intra-community edges means that only the intra-community edges are shown in the layout (represented by "Intra" in, e.g., *CNO (Infomap, RN, RN) Intra*). Similarly for inter-community filtering (represented by "Inter"). The average edge size with only inter connections are high (33.79 for *CNO (Infomap, RN, RN) Inter* and 26.7 for *CNO (Louvain, RN, RN) Inter*), which increases the average edge size of the network when considering all edges (Table 6).

Since a network community is composed by nodes that interact more between them-

selves than with nodes outside the community and most edges are inside the community, we assume that the inter-community edges are relatively less relevant. Considering that the best node reordering algorithm according to the quantitative analysis may not represent the best visual user experience, we also performed a visual analysis. Figure 34 shows two days of the Hospital network using different node reordering strategies: (1) *Appearance*; (2) *RN*; (3) *CNO (Infomap, RN, RN)*; (4) *CNO (Louvain, RN, RN)*; (5) *CNO (Infomap, RN, RN) Intra*; (6) *CNO (Louvain, RN, RN) Intra*. The *Appearance* layout has the most visual clutter amongst all strategies. Strategies 2, 3, and 4 generate similar layouts in terms of visual clutter, however strategy 4 is slightly worse, which is in agreement with the results shown in Table 6.

Inter-community edges increase visual clutter and impair the layout. Although inter-community edges may also have relevant temporal patterns, we focus on the analysis of intra-community edges, which is more challenging. It is not possible to add the *Recurrent Neighbors* method (strategy 2) in the comparative analysis because it does not take into account the community structure. The layout generated by strategy 6 is cleaner than the one generated by strategy 5, allowing the user to identify patterns more easily since visual clutter due to overlapping edges is reduced (Figure 34). Moreover, it allows the study of specific nodes' edges and encourages the user to further explore the layout. Given the coherence between the quantitative and visual analysis, the experiments on the Hospital network will use the CNO configuration *CNO (Louvain, RN, RN) Intra* from now on.

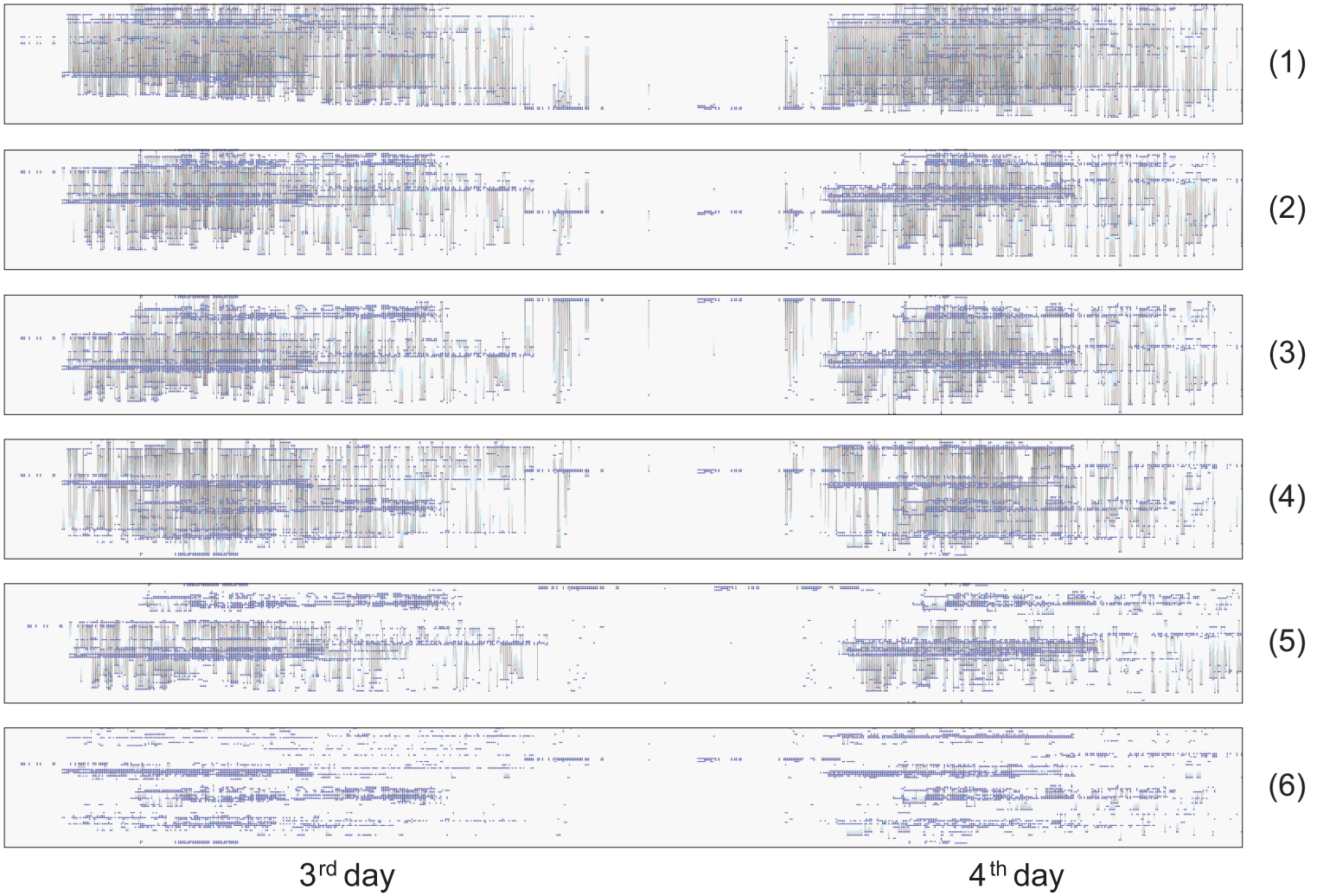

Figure 34 – An overview of the Hospital network considering two of the five days. Layouts were generated using (1) Appearance; (2) RN; (3) CNO (Infomap, RN, RN); (4) CNO (Louvain, RN, RN); (5) CNO (Infomap, RN, RN) Intra; (6) CNO (Louvain, RN, RN) Intra. Each timestamp in the layout refers to a three minute interval and considers all connections that occurred in it. Reprinted from (LINHARES et al., 2019b) ©2019 Elsevier.

Considering the *Recurrent Neighbors* method ordering, Fig. 35 presents the interactions between a patient (node 1383), a medical doctor (node 1157) and a nurse (node 1114) in the first three days in the network. A node is positioned in the line when it is active at the respective timestamp. Edges are shown only between the three selected nodes. As a result, these nodes appear even when there is no connection with the others in selection. In the first day, although the three nodes are active in the network, there is no interaction between them, since there are no edges between the selected nodes. On the second day, it is possible to visualize several interactions between the nodes, being the first one between the nurse and patient (the edge between nodes 1383 and 1114). Furthermore, the nurse made contact with the doctor and return to the patient, from which we can suppose that the patient had some complications and the nurse enter in contact with the doctor to try to solve the problem. Additionally, on the second day, after the contact between the nurse and the patient, the doctor made his first contact directly to the patient. This contact was extended along the third day, indicating a possible treatment of the patient. Understanding the sequence of connections between people in a Hospital may be important, as for example, to track the spread of disease, allowing to adopt appropriate containment measures.

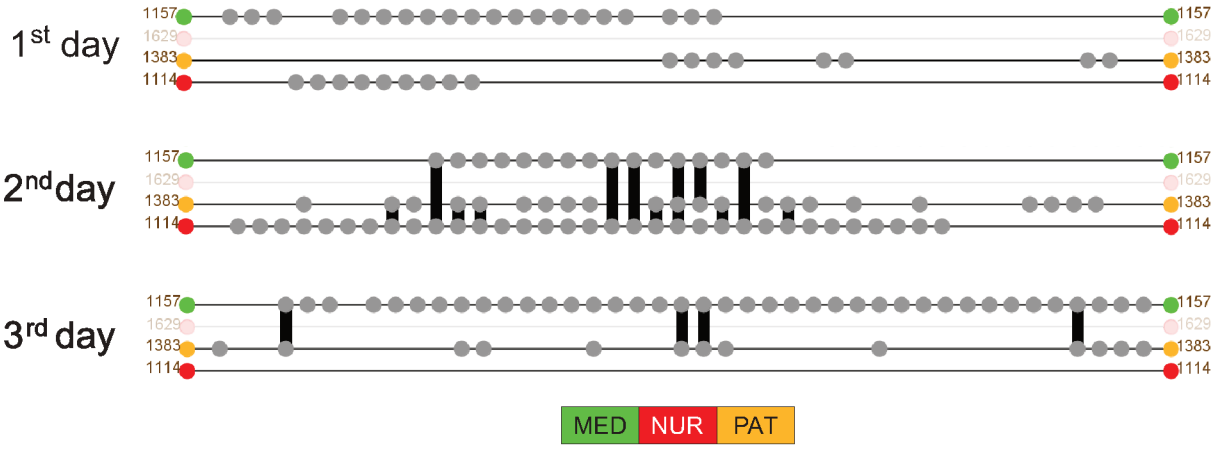

Figure 35 – MSV layout demonstrating the relation between the doctor, nurse and patient in three days of the network. Each timestamp in the layout refers to a 15 minutes interval and considers all connections that occurred in it. Adapted from (LINHARES et al., 2017a) ©2017 SBC.

Figure 36 illustrates how the proximity of nodes that belong to the same community facilitates the analysis of patterns. It is possible to see edges between three nodes of the NUR profile (nurses and nurses' aides). In the first and second days, only the nurse 1625 was active (interacted) in the network. Such interactions are shown as gray circles at specific times. As there are no edges associated with these nodes, the interactions involve others in the network. In the third day, the nurse 1629 had no interactions (it was not active in the network this day) whereas nurses 1295 and 1625 had many

interactions, but not between themselves and neither simultaneously (due to different work shifts (VANHEMS et al., 2013)). In the fourth day, the nurse 1629 also had no connections, while the others had several interactions between themselves. This behavior is shown by edges connecting both nodes during the day, which suggests teamwork. This behavior can also be observed in the fifth day, but with the absence of the nurse 1295 and several connections between the other two nurses. Overall, two patterns can be observed: (1) only the nurse 1625 was active in the network in all five days; (2) the nurses 1629 and 1295 do not connect in the same day, which may be related to their days off.

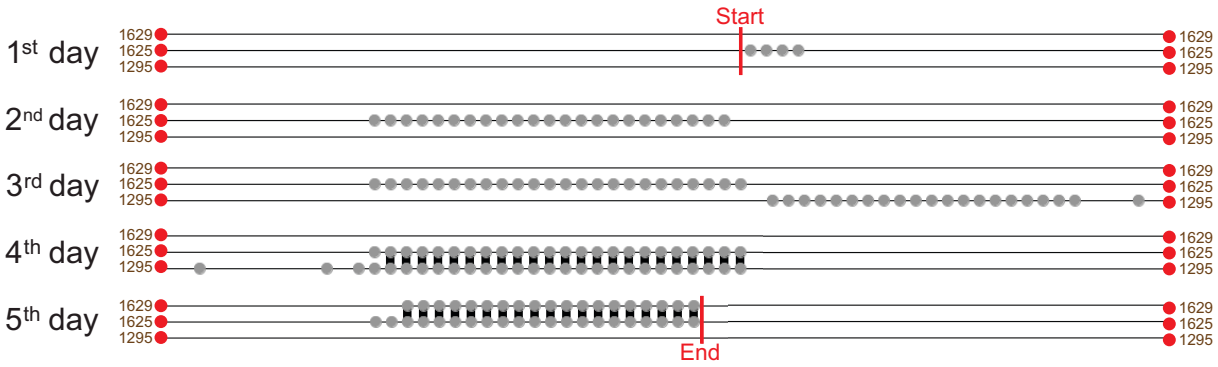

Figure 36 – Interactions between three NURs during the five days of the Hospital network. The network begins in the "Start" bar and finishes in the "End" bar. Each timestamp in the layout refers to a twenty minute interval and considers all connections that occurred in it. Time scale per day: 12:00 a.m. to 11:59 p.m. Reprinted from (LINHARES et al., 2019b) ©2019 Elsevier.

Figure 37 reveals another non-trivial pattern, where two physicians (ids 1168 e 1191) take care of a patient (id 1385) at the same time. Figure 37(a) shows all connections between these three individuals in a specific period of time. By analyzing the edges between each pair of nodes (Figure 37(b-d)), it is possible to verify simultaneous edges between all of them. This pattern suggests the following scenarios: (1) they are physicians of different specialties taking care of a patient who demands special attention; (2) as the network refers to a university hospital, it may be a medical student under supervision of a senior physician in contact with the patient.

The layout could be further improved by using other visualization strategies. One possibility is to use a heatmap view of node activity over time (TAM - *Temporal Activity Map*) (GHONIEM et al., 2014; LINHARES et al., 2017b), which improves the identification of certain patterns. Figure 38 presents the TAM layout using four different node reordering methods (*Appearance, Degree*, RN and *CNO (Louvain, RN, RN)*) and nodes from the NUR (red) and MED (green) profiles. It is not straightforward to identify patterns in nodes from the same profile, due to the distance between each other, when adopting *Appearance* and *Degree* (Figure 38(a-b)). RN was able to separate the nodes according to their profiles (Figure 38(c)), facilitating the identification that all

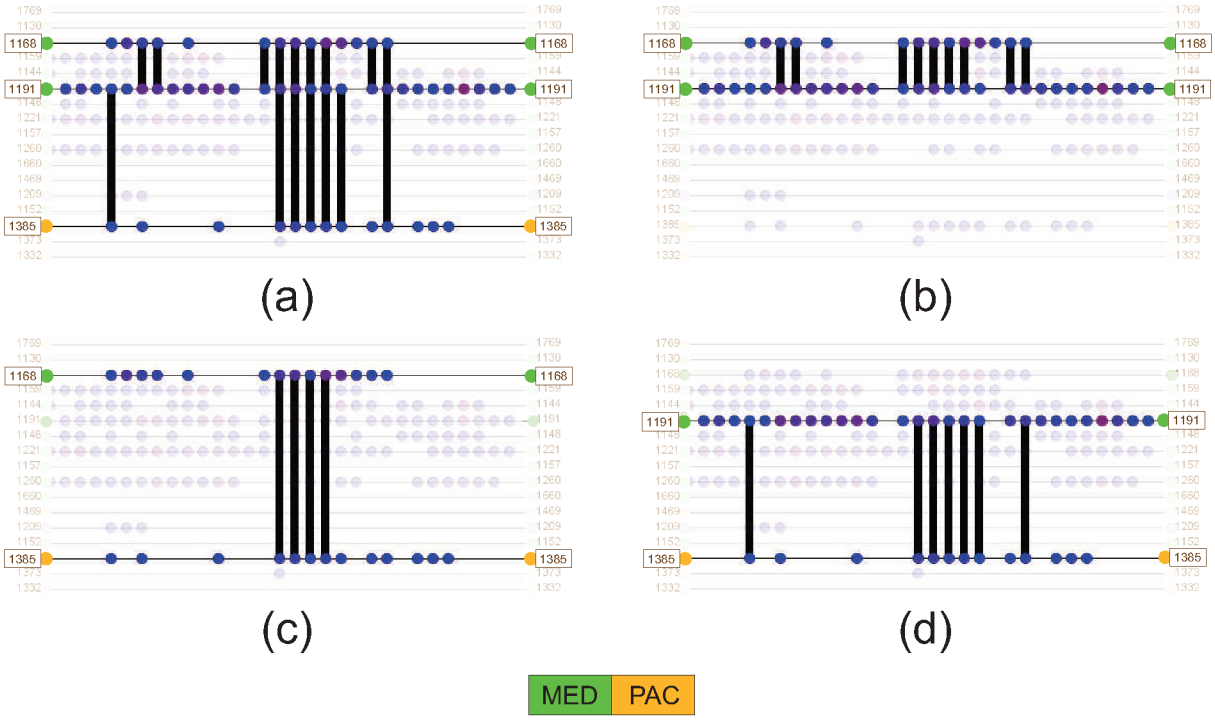

Figure 37 – Simultaneous connections between a patient (PAC profile) and two physicians (MED profile). (a) All connections between the three individuals in a period of time. (b-d) Connections between each pair of individuals for the same time interval from (a). Each timestamp in the layout refers to a 3 minute interval and considers all connections that occurred in it. Reprinted from (LINHARES et al., 2019b) ©2019 Elsevier.

physicians join and leave the network approximately at the same time. However, RN did not position the nodes from the NUR profile as close to each other as possible, making it difficult to identify patterns involving them. Using CNO, the community detection algorithm grouped nurses in a community and the physicians in another, while the re-ordering process positioned them as depicted in Figure 38(d). Besides the well-defined physicians' work shift, also observed with RN, the position of nurses in the CNO layout facilitates the perception that some nurses leave approximately at the same time other nurses join the network. This pattern is expected since there are different work shifts of nurses (VANHEMS et al., 2013).

Figure 38 – Visualization of different layouts using Temporal Activity Map (TAM), for the Hospital network, with focus on profiles NUR (red) and MED (green) during three of the five days: (a) Appearance; (b) Degree; (c) Recurrent Neighbors; (d) CNO (Louvain, RN, RN). Each timestamp in the layout refers to a 30 minute interval and considers all connections that occurred in it. Reprinted from (LINHARES et al., 2019b) ©2019 Elsevier.

4.4.2 Large dataset – Twitter

The Twitter network is a subset of the network presented by Pereira et al. (PEREIRA; AMO; GAMA, 2016) which contains data from retweets that mention a Brazilian newspaper called *Folha de São Paulo* (username @folha). When a user retweets someone's tweet related to news published by this newspaper, an edge is created in the respective time. Such edge occurs between the two users involved in the retweeting process, so these users become nodes in the network. The network comprises a period of 10 days (between July 12, 2016, and July 21, 2016) and is formed by 50,461 nodes and 98,416 edges. As described in Table 8, all connections are classified in topics related to the content of the tweet that each of the edges represents (PEREIRA; AMO; GAMA, 2016). The temporal resolution of the network is one hour, meaning that all connections between two nodes that occur in this interval are considered as one in the layout.

Table 8 – Each topic and the most recurrent subjects in the Twitter network. Adapted from (PEREIRA; AMO; GAMA, 2016).

Topic	Recurrent subjects
Sports	Olympic Games, Eurocup
Celebrity	Luana Piovani
Corruption	Sergio Moro, Lava Jato, Paulo Bernardo
Politics	Impeachment, Temer, Dilma
Education	Science without Borders
Security	Rio de Janeiro
International	Terrorism, Brexit

Table 9 shows a comparative analysis of RN, CNO, and three naive node reordering strategies (*Appearance*, *Lexicographic* and *Degree*). CNO was evaluated using both Louvain and Infomap algorithms (Step 1) and RN in Steps 2 and 3. We focus on RN with CNO because preliminary tests showed that the produced layout had almost twice fewer intersections than *Lexicographic*, which was the best naive strategy. For large and dense networks such as the Twitter network (the Twitter network has approximately 440 edges per time on average in contrast to the Hospital network with six edges per time on average), the analysis may be further improved with a hierarchical approach and thus we apply CNO on Twitter using three different levels. Since the *Appearance* strategy generates the layout with most visual clutter (i.e., more intersections), we take this strategy as a baseline to analyze the improvement provided by other methods.

The hierarchical CNO decomposes the network communities at each iteration. When a community is divided into two or more communities, some intra-community edges now become inter-community edges in the new level. Therefore, adding more levels implies on less intra- and more inter-community edges and thus less edge overlapping in the layout (Table 9). Comparing *CNO (Louvain, RN, RN)* at different levels, the third level contains

Table 9 – Quantitative analysis using different node reordering algorithms for the Twitter network (50,461 nodes and 98,416 edges). For CNO, when there is no indication of which edge filtering is used, all edges are considered in the analysis. Reprinted from (LINHARES et al., 2019b) ©2019 Elsevier.

Node Reordering Algorithm	CNO Level	# Overlapping Edges	Avg Edge Size	# Intersections	Less visual clutter than Appearance (%)
Appearance	—	98,401	13,809.97	137,780,128,036	—
Lexicographic	—	98,415	15,649.44	104,044,404,629	24.49
Degree	—	98,416	15,072.14	112,358,098,034	18.45
Recurrent Neighbors (RN)	—	98,411	8,347.97	56,738,821,369	58.82
CNO (Infomap, RN, RN)	1	96,273	4,673.5	11,782,839,610	91.45
CNO (Infomap, RN, RN) Intra	1	60,561	1,085.32	1,871,868,509	98.64
CNO (Infomap, RN, RN) Inter	1	26,822	14,235.97	8,558,588,022	93.79
CNO (Louvain, RN, RN)	1	95,877	3,131.52	7,164,444,724	94.80
CNO (Louvain, RN, RN) Intra	1	79,605	1,340.24	2,696,896,654	98.04
CNO (Louvain, RN, RN) Inter	1	14,105	13,833.36	2,486,398,635	98.20
CNO (Louvain, RN, RN)	2	95,793	3,126.08	6,849,004,216	95.03
CNO (Louvain, RN, RN) Intra	2	52,347	1,122.52	1,730,911,726	98.74
CNO (Louvain, RN, RN) Inter	2	35,531	6,654.53	4,162,333,350	96.98
CNO (Louvain, RN, RN)	3	95,794	3,127.83	6,825,292,768	95.05
CNO (Louvain, RN, RN) Intra	3	44,633	1,199.41	1,715,433,398	98.75
CNO (Louvain, RN, RN) Inter	3	40,662	5,853.81	4,202,544,529	96.95

almost half of the intra-community edges of the first level (44,633 vs 79,605, respectively).

The Recurrent Neighbors method alone cannot provide an effective layout for the Twitter network analysis due to the network size (Figure 39(a)). Figure 39(b-d) shows that the CNO facilitates the visual analysis by using its hierarchical procedure to filter less relevant edges and focusing on only the intra- or inter-community edges at any level. Figure 39(c-d) shows, for example, that communities at level 1 are decomposed in multiple smaller communities at level 2. Since a number of intra-community edges at level 1 become inter-community edges at level 2, they are not shown when further filtering is applied. The advantage of this method is that the user can study a localized part of the network, as, for example, specific nodes inside a community or compare the activity inside and across the different communities (Figure 39(e)). Note also that, for the Hospital network, the layouts from CNO with different configurations were usually similar to the RN layouts. However, in the Twitter network, RN results are always worse than any CNO configuration for any quantitative measure.

The analysis of large networks may be further improved with the identification of patterns in a global view of the layout combined with our proposed taxonomy. The categorization of these patterns, which represent recurrent node/edge activity and are visually easier to identify, may help to enhance the visual analysis. Based on the results in Table 9 and Figure 39, we performed experiments for the Twitter network using *CNO (Louvain, RN, RN) Intra* to generate and visualize communities, and used our proposed taxonomy to categorize the communities. Table 10 shows the parameters used in the analysis.

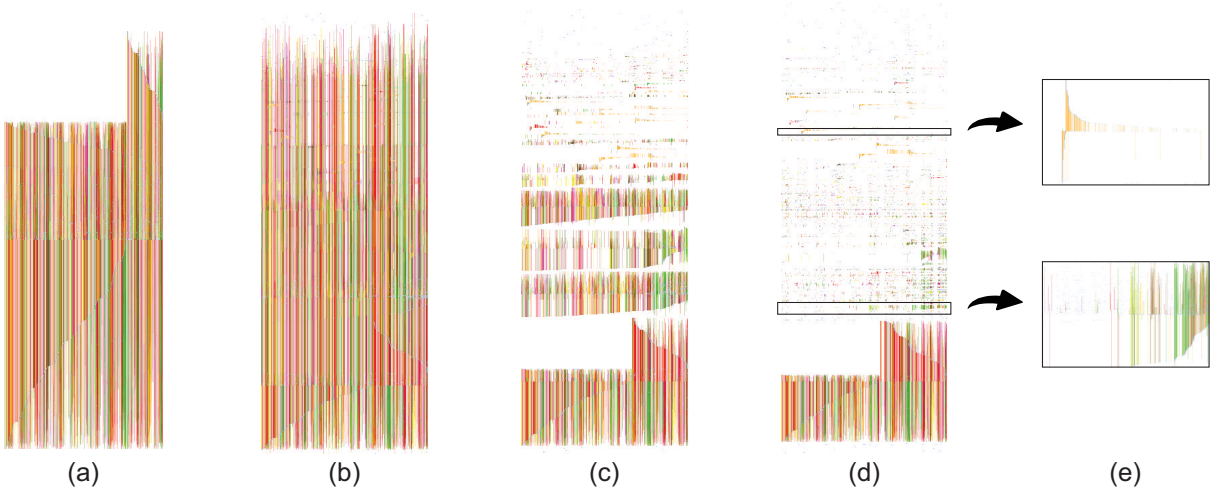

Figure 39 – An overview of the Twitter network using different node reordering strategies: (a) Recurrent Neighbors; (b) CNO (Louvain, RN, RN) level 1; (c) CNO (Louvain, RN, RN) Intra, level 1; (d) CNO (Louvain, RN, RN) Intra, level 2; (e) Examples of network communities in (d), which leads to local analysis. Reprinted from (LINHARES et al., 2019b) ©2019 Elsevier.

Table 10 – Taxonomy parameters and values chosen for the Twitter network. Reprinted from (LINHARES et al., 2019b) ©2019 Elsevier.

Parameter	Value	Value based on
$T_{\text{Final}} - T_{\text{Initial}}$	224	Amount of timestamps in the network
S_I	17.86	Global average considering the time interval between two consecutive connections in each community
T_{dg}	1	The most frequent median time interval between two consecutive connections (mode of the medians)
T_h	90%	Empirical analysis

The analysis of the Twitter network using CNO level 3 resulted in 3,918 non-overlapping communities at this level, with an average size of 12.82 nodes. Table 11 shows the percentage of network communities classified in each category of the taxonomy. The values found may be related to network semantics. The average number of intra-edges per community is 9.49. This relatively small value makes the existence of communities with continuous or grouped activity more difficult as the probability for such edges to be dispersed over time is higher.

To demonstrate the diversity of community types in the Twitter network, we perform a visual analysis. Figure 40 shows four communities categorized as homogeneous; each exemplar community has most edges related to a single topic, represented by the assigned colors. Moreover, Figure 40(a) is classified as having continuous activity whereas Fig-

Table 11 – Community classification in each category of the taxonomy for the Twitter network. Reprinted from (LINHARES et al., 2019b) ©2019 Elsevier.

Category	Occurrence (%)
Sporadic Activity	99.45
Continuous Activity	0.55
Dispersed Activity	93.74
Grouped Activity	6.26
Homogeneity	52.32
Heterogeneity	47.68

ure 40(b,c,d) have sporadic activity (being the ones from Figure 40(b,d) of the initial type). In terms of activity dispersion, Figure 40(a,b,d) have grouped activity whereas Figure 40(c) is dispersed.

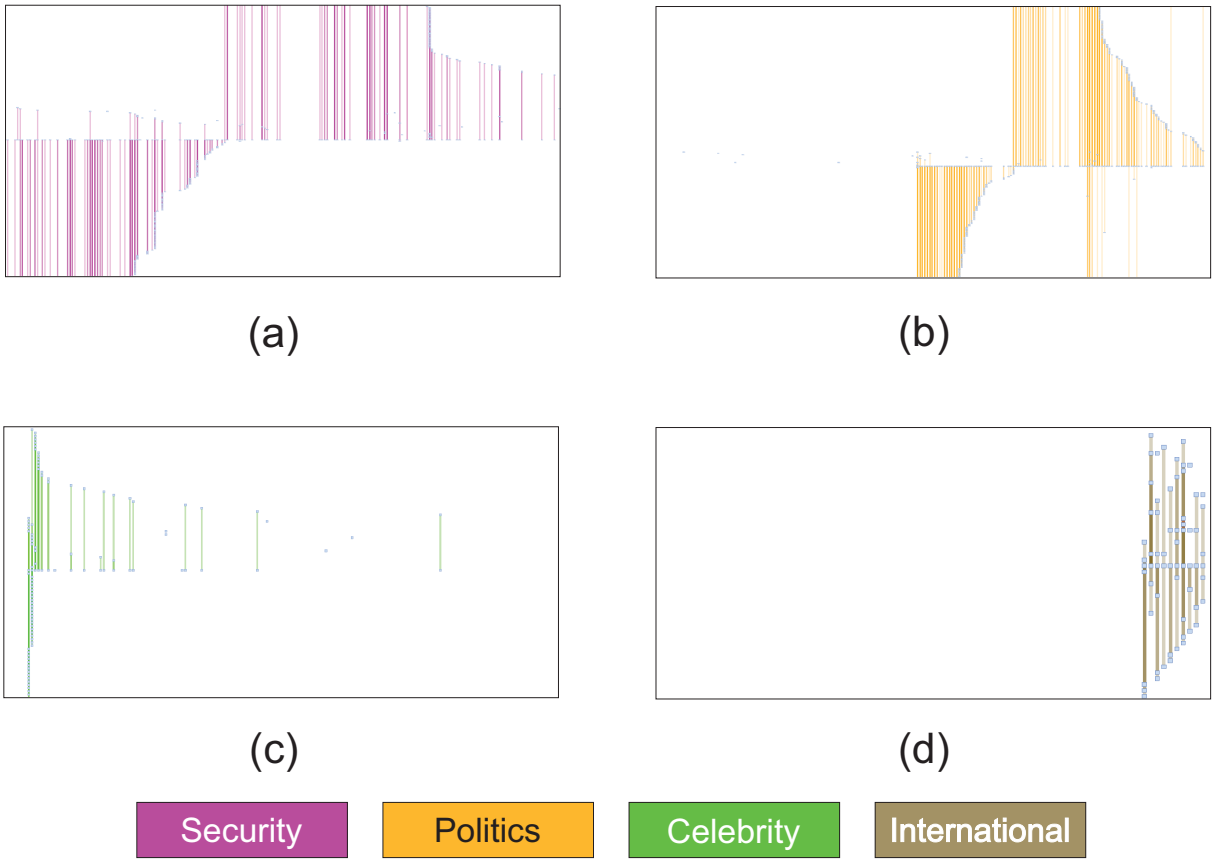

Figure 40 – Four homogeneous communities from Twitter network visualized using CNO level 3. The community in (a) is classified as having continuous activity, while (b,c,d) have sporadic activity; (a,b,d) have grouped activity, while (c) is dispersed. Reprinted from (LINHARES et al., 2019b) ©2019 Elsevier.

The identification of the homogeneity is useful to detect groups of individuals that share a common interest (KAPANIPATHI et al., 2014; LIM; DATTA, 2012). For exam-

ple, a business manager may use this information to understand the user preference and personalize product recommendation. Moreover, the manager can act according to the community activity pattern: if the manager identifies that the individuals in one community talk about a specific product only during a few days (i.e., sporadic activity) or talk about the product every day but a few times per day (i.e., continuous but dispersed activity), he or she can adapt the marketing strategy accordingly. In this example, CNO helps the manager to identify periods of time to optimize marketing campaigns.

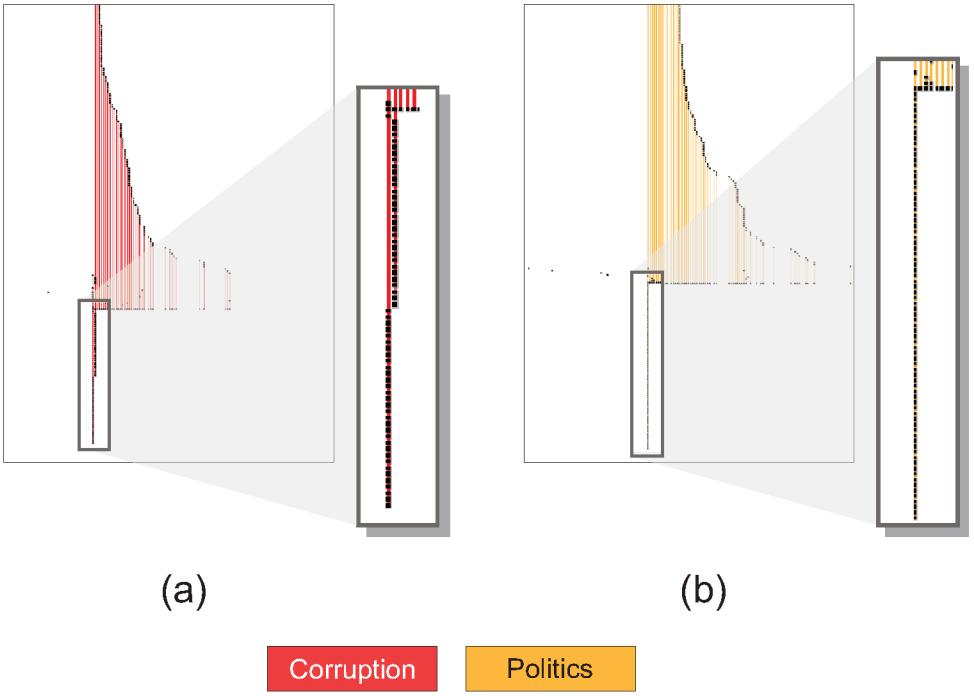

Figure 41 – Two homogeneous communities visualized using CNO with intra-community edges in the Twitter network. Panels (a) and (b) show a spike in the highlighted time. This pattern suggests a "bombshell" tweet since it was retweeted a lot in a single time. Reprinted from (LINHARES et al., 2019b) ©2019 Elsevier.

Some of the network communities have a specific visual pattern: a spike of edges at a particular time. Figure 41 shows a situation where several nodes have connections in specific times with intra-community activity. The spike suggests the occurrence of a "bombshell" tweet as it was retweeted a lot in a single time (IMRAN et al., 2014). This spike represents edges between the focal node and all the others that connect to it in this time (each one of these nodes is visually represented by a black dot). The focal node is drawn in the center of the layout due to the chosen node reordering strategy (*Recurrent Neighbors*), which centralizes the most connected nodes. We assume this central node as an influential person in its community since it is in most of the intra-community edges. Such influencer is also seen in most of the communities shown in Figure 40. In fact, this pattern can be seen in almost all communities detected in our Twitter network. In this

case, the hypothetical business manager could invite the influencer to act in marketing campaigns.

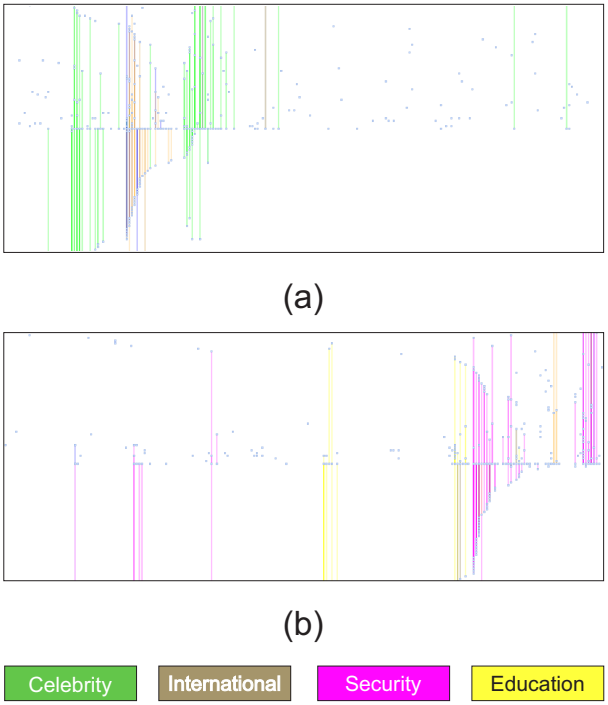

Figure 42 – Two heterogeneous communities with topic change visualized using CNO level 3. Recurrent topics (a) "Celebrity" and (b) "Security" are discussed in alternating time intervals. Reprinted from (LINHARES et al., 2019b) ©2019 Elsevier.

Another pattern observed in the Twitter network is a topic change in some of the heterogeneous communities (Figure 42). Figure 42(a) shows that an initial block of retweets about the topic "Celebrity" is replaced by other topics and then return to the first topic. The same dynamic occurs in Figure 42(b), in which the topic "Security" is discussed in the first time interval, then replaced by the topic "Education" in the second interval and so on. The identification of topic swaps may be used to study changes in user behaviors within a group, allowing to identify, for example, the consumption of seasonal products (SHARMA, 2014; CARRASCOSA et al., 2013) or the connection between related topics.

In several of the observed heterogeneous communities, it is difficult to identify blocks of consecutive times with edges related to a common topic, as in the examples of Figure 42. Figure 43 illustrates cases in which the topic change is so frequent that such blocks may not exist, a situation in which each time refers to a different topic (see, for example, Figure 43(a)). The community from Figure 43(a) is also sporadic and dispersed whereas Figure 43(b-d) have continuous and grouped activity. Besides the community categorization, other filters may be applied to guide the user in the analysis of these communities. One effective possibility, for example, is to allow the user to focus on specific

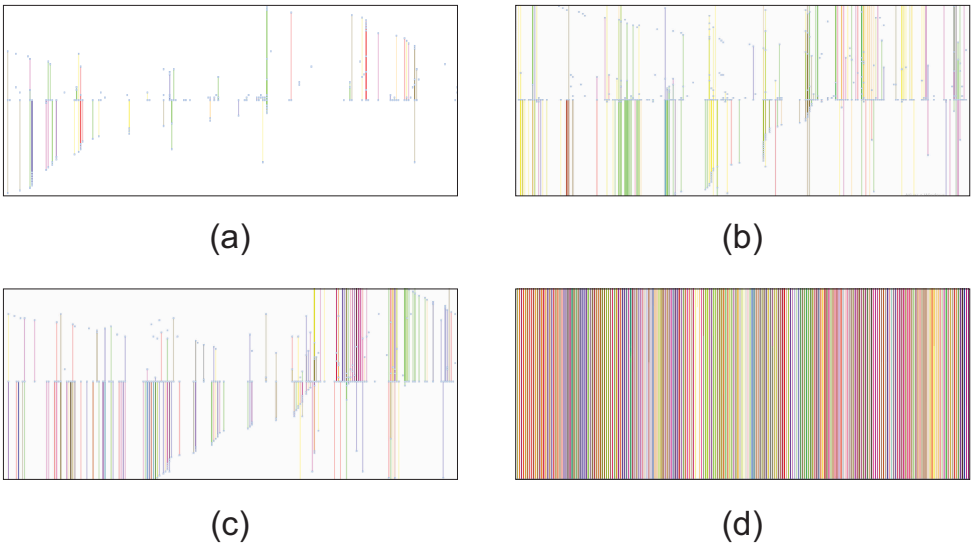

Figure 43 – Four heterogeneous communities visualized using CNO level 3 with different levels of activity. The community in (a) is sporadic and dispersed, while (b,c,d) have continuous and grouped activity. Reprinted from (LINHARES et al., 2019b) ©2019 Elsevier.

communities according to their levels of activity, i.e., the percentage of distinct times that have any edge activity. Figure 43 illustrates such different levels of activity when comparing the communities: Figure 43(c) has 51.57% of activity whereas the community from Figure 43(d) has 100% of activity (i.e., have edges at all times).

4.5 Limitations

The *Community-based Node Ordering (CNO)* method provides means to reduce the visual complexity of large networks by exploiting its own structure yet allowing the parallel analysis of multiple nodes. The methodology emphasizes patterns that could not be identified by simply zooming in/out or by using only node reordering algorithms. Although CNO substantially reduces visual clutter and improves the visual analysis of large temporal networks, it has limitations when applied to very large networks. There is also a compromise between removal of information (i.e., removing edges or breaking down in further levels) and significance of the remaining information (i.e., the visualized edges and communities). Nevertheless, this approach allows the user to interactively identify the appropriate level of cleaning according to the context.

The main issue, however, is that some networks have a weak community structure or have a community resolution limit that forbids advancing to higher hierarchical levels to further break down the communities. In the absence of community structure, for example in the worst-case scenario of random networks, our method is expected to perform no worse than random placement of nodes in the temporal layout. However, since community detection algorithms are based on optimization processes, community structure is

always detected in a single instance of a random network (even if this is meaningless for a purely random network). In this case, from first principles, our method always performs equal or better than random placement of nodes in the layout. The resolution limit, however, cannot be avoided but this is a limitation inherent of network community detection algorithms. Our method is optimized to non-overlapping communities. Although this is a limitation, further amendments in our method could be done, for example by replicating overlapping nodes in the overlapping communities to not compromise the information lost when removing inter-community edges.

Furthermore, when analyzing temporal networks one should be aware of potential biases generated by sampling design. In particular, the temporal resolution and the period of observation (e.g., 1 day or 1 week) may affect the visual information. This is not, however, a limitation of our method but an intrinsic characteristic of temporal networks (ROCHA; MASUDA; HOLME, 2017). Our method accommodates such limitations but naturally the visual information will be different according to different choices of temporal parameters. Not least, another problem is that network communities are detected using the static version of the network where edges are aggregated over the entire observation period. Although network communities are often stable, they may sometimes vary in time. From the visual analysis point of view, this is a paradoxical limitation because changing communities would increase the visual complexity, since nodes would change positions in time, and consequently pollute the layout. Furthermore, changing communities would likely show new activity patterns and call for further classes in our taxonomy. We thus adopted a conservative strategy and made the simple assumption of community stability as a first attempt to improve the analysis. Also, for the taxonomy community categorization, unbalanced community sizes, i.e., number of nodes inside each community, can create a tendency in the classification and generate bias results. Further investigation in this subject is required.

In large communities with a high level of node activity (e.g., Figure 43(d)), the edges between two nodes that are spatially distant generate long vertical lines. In this case, a new hierarchical level could be used (via interactive tools) to the next CNO level. The result may be a layout that focuses on a more detailed perspective of the community with less visual clutter. Another possibility is to set CNO using a different configuration, in an attempt to approximate these nodes through the new node ordering. However, finding the best CNO configuration is not a trivial task. The choice of the optimal methods to detect network communities and to reorder nodes, the number of hierarchical levels, or which edge filtering to apply (intra- or inter-community), highly depends on the user task and requests an initial exploratory analysis.

4.6 Final considerations

We have introduced a novel and visual scalable node reordering approach called *Community-based Node Ordering (CNO)* to perform visual analysis of temporal networks. Our method improves the visual analysis of large temporal networks by hierarchically breaking down communities and removing less relevant edges at each level. A novel taxonomy was also proposed to categorize communities according to the activity patterns of their nodes and edges using the CNO layout, helping the user to study larger temporal networks and more efficiently compare groups of nodes. We performed a series of quantitative analysis and visual explorations using small and large real-world networks to demonstrate the superior performance of the CNO layout in comparison to existing node reordering strategies, validating our Hypothesis. The results demonstrated coherence between the visual and quantitative analysis and allowed the identification of patterns that would be difficult to see without CNO, for example, work shifts in the Hospital network and the so-called "bombshells" in the Twitter network. The CNO edge filtering mechanism, based on network communities, generally improves the visual analysis of temporal networks but is particularly suitable for larger and denser networks. Although the method has its limitations when it comes to non-stable communities or too large networks, it is a promising first step to improve visualization of massive real-world networks.

Chapter 5

Visualization of dynamic processes on temporal networks

This chapter describes the visualization of dynamic processes applied to temporal networks. This visualization helps to reveal nontrivial random walker trajectories, allowed mapping of infection transmission paths and the identification of timings and directions of infection events. This chapter is based on the published book chapter (LINHARES et al., 2019a). Reprinted/adapted by permission from Springer Nature: Springer, Visualisation of Structure and Processes on Temporal Networks by Claudio D. G. Linhares, Jean R. Ponciano, Jose Gustavo S. Paiva et al © 2019. The final authenticated version is available online at: <http://dx.doi.org/10.1007/978-3-030-23495-9_5>.

5.1 Visual analysis

A useful application of temporal visualization is the analysis of dynamic processes on networks (MASUDA; LAMBIOTTE, 2016). For example, the evolution of a diffusion process can be monitored by coloring nodes according to their dynamic state or by coloring edges to highlight specific paths. Two fundamental dynamic processes of theoretical and practical interest are the random walk and the infection (information) spread dynamics. In the random walk model, a node i can be in one out of two states at each time step τ, i.e. occupied $\phi_i(\tau) = O$ or empty $\phi_i(\tau) = E$. All network nodes but one randomly chosen node i_0 start empty at time t_0. At each τ, the walker decides whether to move via an existing active edge to a neighbor with probability $(1 - p)$ or to remain in the current node with probability p. If there are no active neighbors at τ, the walker simply remains in the node (STARNINI et al., 2012; ROCHA; MASUDA, 2014). The process unfolds until $t_f = T$.

We use four infection models, SI, SIS, SIR, and SIRS, which the initials represent the name of the possible states, as illustrated in Fig. 44. The SI and SIS, i.e., a two states (susceptible, infected) models, a node i can be in one out of two states at each time step

τ, susceptible $\phi_i(\tau) = S$ or infected $\phi_i(\tau) = I$. In the case of SI, an infected node remains infected in all remaining time steps (Fig. 44(a)). In the SIS model, the infected node remains infected for t_r time steps and then are susceptible again (Fig. 44(b)). In the SIR model, i.e., a three states model (susceptible, infected, recovered), a node i can be in one out of three states at each time step τ, susceptible $\phi_i(\tau) = S$, infected $\phi_i(\tau) = I$, or recovered $\phi_i(\tau) = R$ (Fig. 44(c)). In this case, all nodes start susceptible and one random node is chosen to be initially infected i_0 at time t_0 (patient zero or seed). In case of active neighbors at time t, an infected node i infects a neighbor j with probability β. An infected node remains infected for t_r time steps and then recovers. Recovered nodes cannot be re-infected or turn back to susceptible state (BARRAT; BARTHéLEMY; VESPIGNANI, 2008; ROCHA; BLONDEL, 2013). Another infection model is SIRS, which has the same particularities them SIR, with the difference that after recovered state, the node i has a probability of immunity loss ϵ, which turns back to susceptible state (Fig. 44(d)) (BARRAT; BARTHéLEMY; VESPIGNANI, 2008; ROCHA; BLONDEL, 2013). To implement the infection we use the reactive process (RP), i.e., for each individual infected, there is a probability to infect all of his contacts at the next time stamp (GóMEZ et al., 2010).

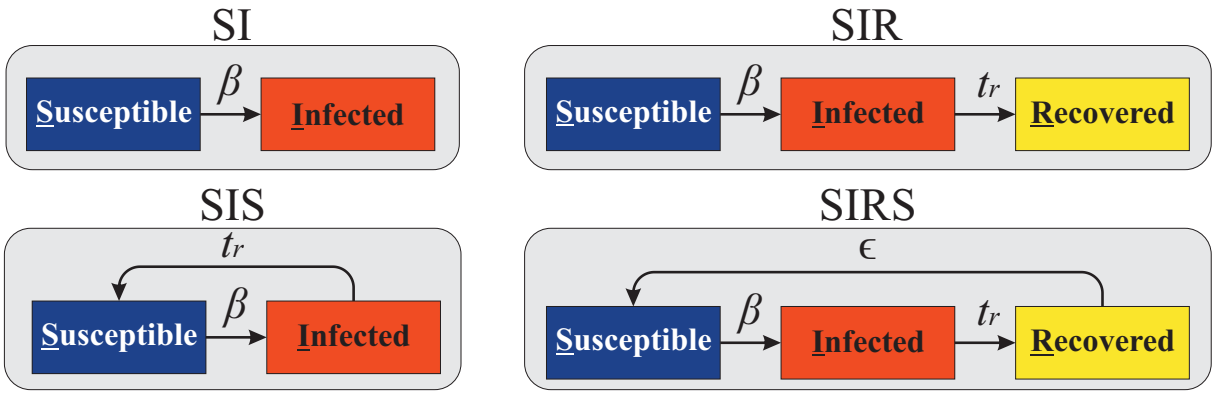

Figure 44 – Four infection models: SI, SIS, SIR, and SIRS. The letters of the model names represent the possible states.

A museum data set of social contacts will be used to illustrate the application of the methods. The network were collected using RFID proximity sensors in the Science Gallery in Dublin, Ireland, during 1 day in 2009, containing 72 visitors and 6980 interaction among them (ISELLA et al., 2011). Figure 45 shows, in a TAM layout, various trajectories of a random walker starting from the same node in the museum network (seed i_0) but using different probabilities p of staying in the node. In the trivial case, $p = 1$ and the walker simply remains in the initial node i_0 indefinitely. The occupancy of this node by the walker is thus seen until the last activation of the hosting node, that happens well before the end of the observation time T (Fig. 45(a)). However, for $p \neq 1$, the walker hops between nodes through active edges and a richer diffusion dynamics unfolds in time (Figs. 45(b-d)). Note that $p = 0$ implies that a walker hops as soon as an active edge

becomes available and less hoping (and simpler trajectories) is observed for larger p, as expected. In these particular random cases, the walker is able to reach longer times for $p \neq 1$ in comparison to $p = 1$, and ends up in nodes that entered later in the network (compare last appearance of the walker for different p in Fig. 45). For any $p \neq 1$, the walker remains trapped between two nodes for relatively long periods, an unlikely scenario in static networks where walkers explore larger regions of the network (STARNINI et al., 2012). Since nodes are ordered by time of first activation, one can also identify potential correlations between lifespan and frequency of walker visits. Similar visualization ideas could be applied to trace higher-order random walks on temporal networks (SCHOLTES et al., 2014).

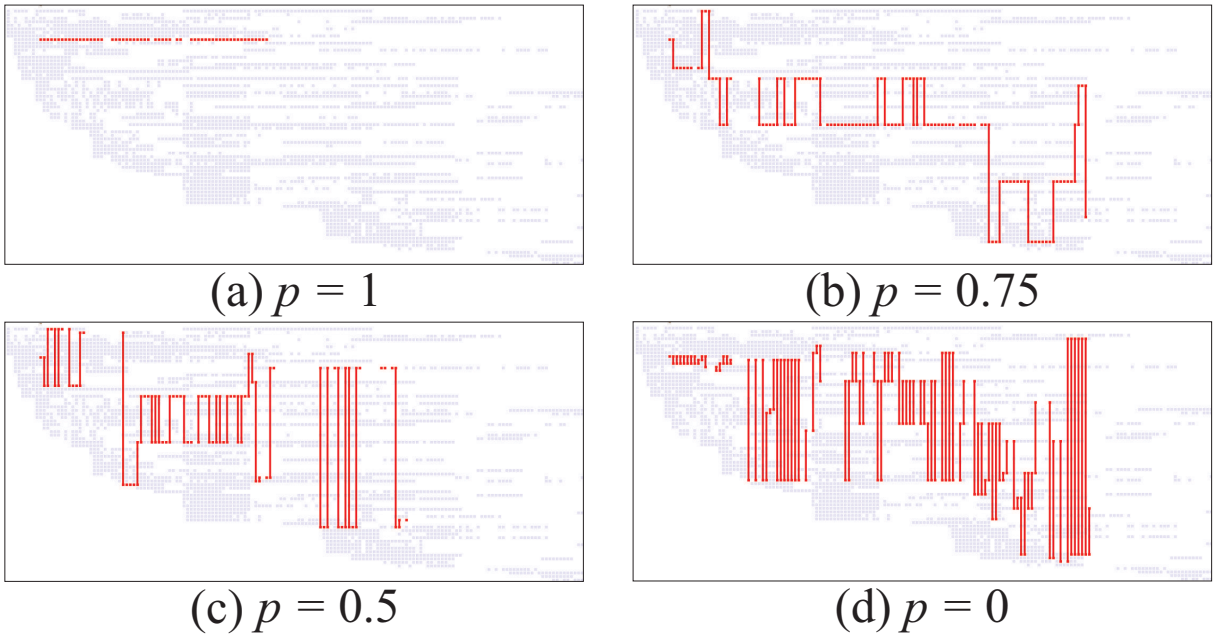

Figure 45 – TAM layout for visualization of dynamical process showing random walk trajectories for the museum contact network with $\delta = 2$ min. The color red indicates the nodes occupied by the walker and the edges used to move between nodes at given times. Light gray indicates unoccupied nodes and white indicates absence of activity. Nodes are sorted by order of appearance. The seed is the same for all cases. Adapted from (LINHARES et al., 2019a) ©2019 Springer Nature Switzerland AG.

In more elaborated dynamic processes as the infection dynamics, a visualization may focus on the state of nodes or on transmission paths, or even both but then likely causing cognitive overload (HUANG; EADESBAND; HONG, 2009). The TAM layout emphasises the node state and is particularly helpful to identify the timings of infection events and how groups of same-state nodes evolve. Figure 46 shows the evolution of the state of all network nodes (for the museum data set) in the SIR infection dynamics for various values of β and t_r. In the trivial case ($\beta = 0$), the infection seed becomes active more than one before turning to the recovery (yellow) state (Fig. 46(a)). In the worst case scenario

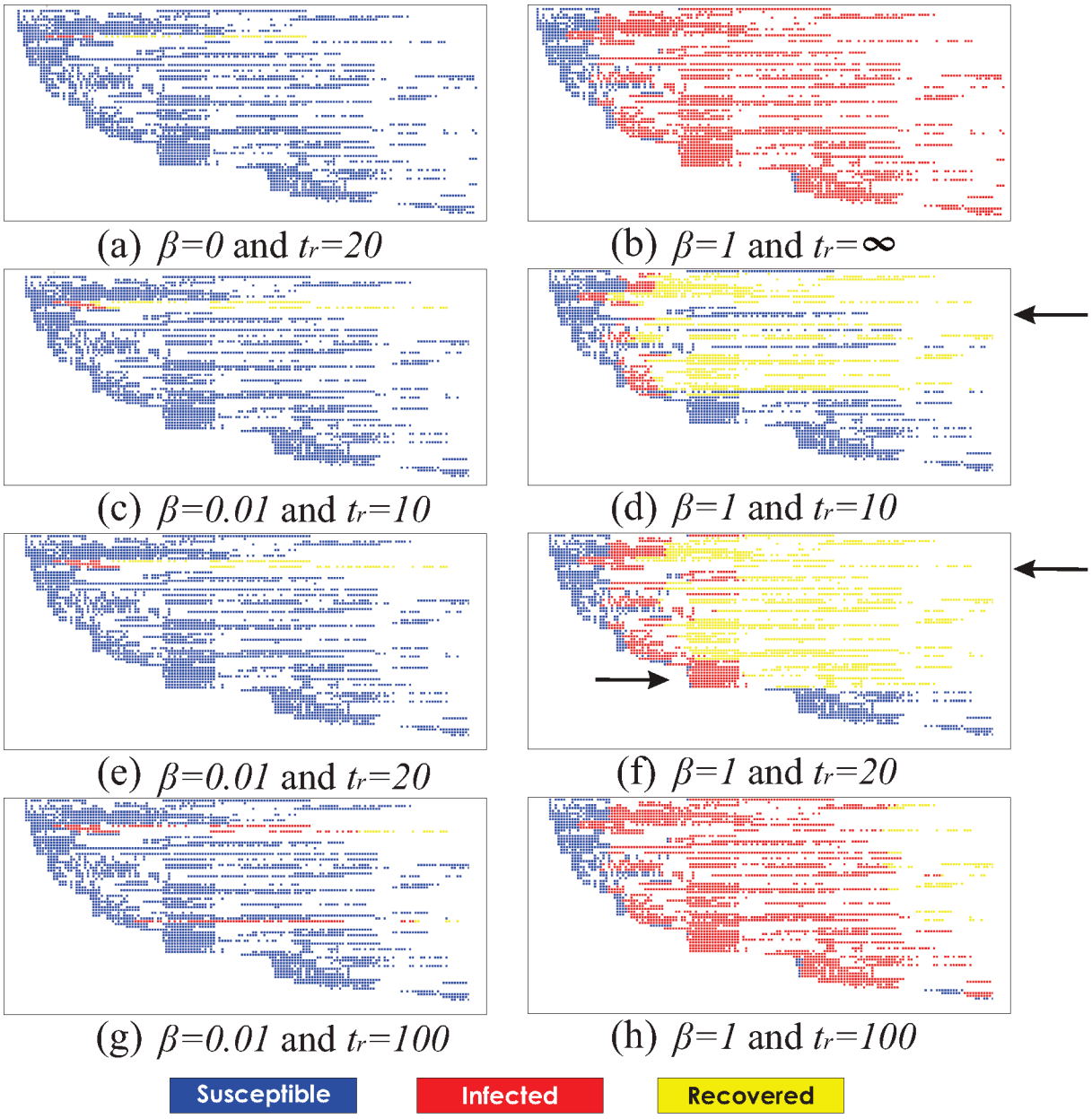

Figure 46 – Infection dynamics for the museum contact network with $\delta = 2$ min. Bottom arrows in (e,f) indicate a temporal cluster of active infected nodes, and top arrows in (d,f) indicate that one node was active and susceptible before the epidemic outbreak. Nodes are sorted by order of appearance. The infection seed is the same for all cases. Adapted from (LINHARES et al., 2019a) ©2019 Springer Nature Switzerland AG.

($\beta = 1$ and $t_r = \infty$, Fig. 46(b)), some nodes remain susceptible for a while after joining the network however everyone eventually becomes infected in this particular network configuration. Overall, this layout facilitates a global understanding of the impact of certain parameters in the dynamics. One may explore variations in the infection period t_r when β is fixed (Figs. 46(c,e,g) or Figs. 46(d,f,h)) or fix t_r and study the infection probability β (Figs. 46(c,d), Figs. 46(e,f) or (Figs. 46(g,h)). For example, a small increase

in β or t_r generated a temporal cluster of active infected nodes (see bottom arrow in Figs. 46(f)). In the case of $\beta = 1$, one node (see top arrows in Fig. 46(d,f)) was active and susceptible before the epidemic outbreak and remained active and susceptible after the end of the outbreak.

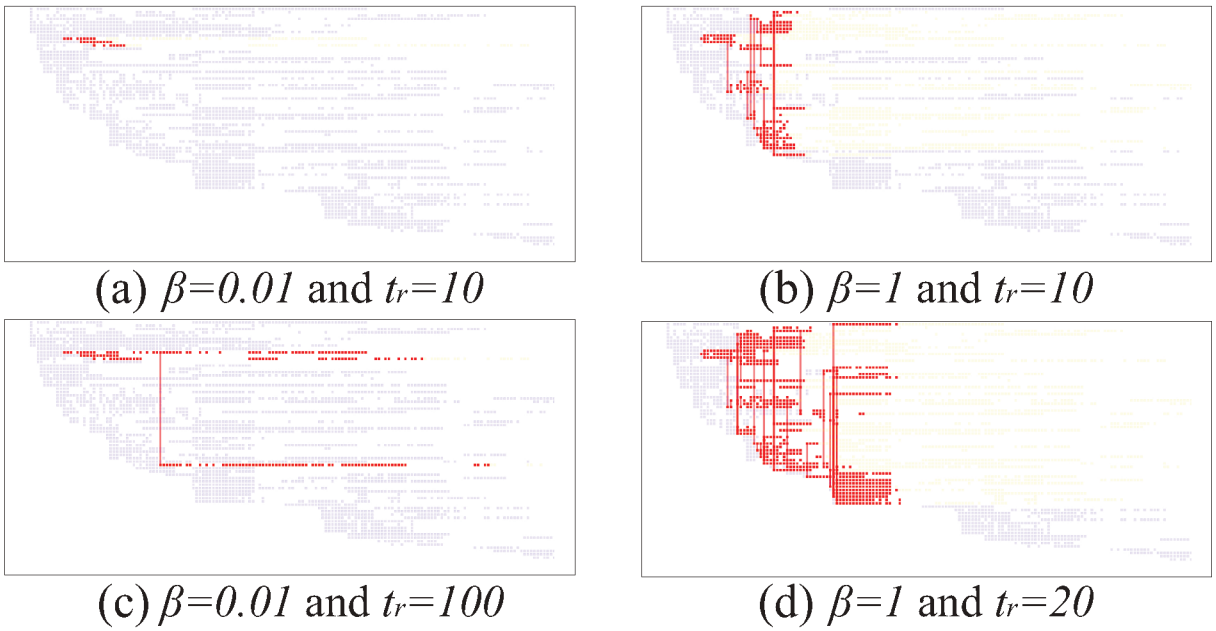

Figure 47 – Infection dynamics for the museum contact network with $\delta = 2$ min. Nodes are sorted by order of appearance. The infection seed is the same for all cases, and the same as for Fig. 46. Adapted from (LINHARES et al., 2019a) ©2019 Springer Nature Switzerland AG.

The limitation of the previous layout is that information of the transmission paths, i.e. the edges through which infection events occurred, is unavailable. Those edges can be included to create a layout of transmission trees, i.e. the sequence of nodes and edges in which the infection (or information) propagates from the seed, with transparency used on non-infected nodes to reduce information load yet allowing a global view of the nodes' states (see Fig. 47 and compare with Fig. 46(c,d,f,g) as they use the same parameters). This new layout helps to identify who infected whom and the timings of these infection events. It is particularly useful to visualize the importance of certain nodes and edges to regulate the infection spread, for example, by visualizing the transmission trees before and after vaccination, i.e. removal of those nodes or edges. The intensity of edge color is also used to identify edge overlap but since infection events are rarer than the chance of having a contact in a given time step, overlap (and thus visual clutter) is less of a problem than if all active edges were shown. The layout can be further optimised by ordering nodes according to timings of infection, with earlier infected nodes positioned on the central part of the layout and those infected later at more peripheral positions. This is similar to the RN algorithm however using timings of infection events for ordering

nodes.

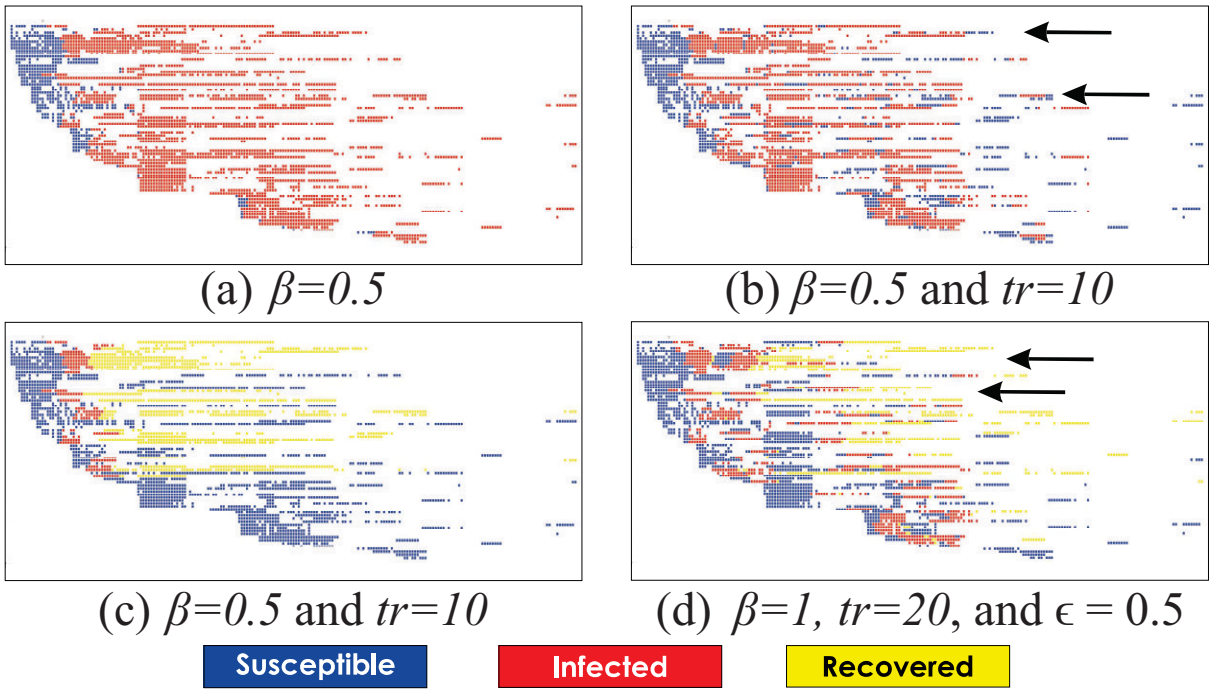

Figure 48 – Infection dynamics for the models (a) SI, (b) SIS, (c) SIR, and (d) SIRS, for the museum contact network with $\delta = 2$ min. Nodes are sorted by order of appearance. The infection seed is the same for all cases.

Besides the SIR infection model, there are other models to simulate the disease spreading. Figure 48 illustrates the difference between the four mentioned infection models, all using the same probabilities, time of recovery and immunity loss. The particularity of each model is reflected in the visualization, for instance, in Figure 48(a-b) are respectively represented the SI and SIS, which do not present the recovered state. Figure 48(c-d) presents the SIR and SIRS models, respectively. Notice that are several occurrences of nodes turning back to susceptible (see arrows in Figs. 48(b,d)), which creates complex layouts of transmissions trees.

5.2 Visual and computational limitations

In the temporal layout, node re-ordering methods optimise the distribution of edges, reducing edge overlap (visual clutter) and improving the visualization. Nevertheless, two-dimensional spatial constrains also create lengthy edges crossing several in-between nodes and hindering relevant structural information. If activity is relatively high (e.g. a couple of edges active at the same time step) and the network is large (some hundreds of nodes), re-ordering techniques may be insufficient to provide meaningful visualization (KEIM, 2001). Alternative solutions under development include identifying and removing specific edges,

e.g. those edges between or within network communities. Moving to 3D layouts (with two-dimensions for space and one-dimension for time) may also improve visualization by disentangling overlapping edges at the cost of more information being available. The temporal activity map solution is more scalable with network size and density since it removes the issue of edges overlap. Combined with node re-ordering techniques to improve the placement of nodes, this solution may highlight relevant activity and structural patterns with thousands of nodes and any edge density (without showing the edges). The main advantage is the possibility to embed information about the network structure, dynamic processes or meta-data in each node through a coloring scheme with low computational cost.

A crucial step when studying temporal networks is the choice of the observation period (T) and the temporal resolution (δ). Both parameters affect the visualization since the viewer is effectively looking at a static network on screen. The observation period is in principle less critical since one can zoom in/out or move the network around but since some re-ordering algorithms are based on the cumulative network measures, this period may affect the location of nodes (e.g. longer periods may imply on more edges and thus higher strength (ROCHA; MASUDA; HOLME, 2017)). On the other hand, variations in the temporal resolution may substantially change the network structure (RIBEIRO; PERRA; BARONCHELLI, 2013; ROCHA; MASUDA; HOLME, 2017) and thus the information (network structure) being viewed. Although node re-ordering algorithms work for any resolution and easily accommodate all these cases, the viewer has to keep in mind the potential limitations or variations in the structure and activity when performing qualitative visual network analysis.

Since the visualization per se is static, the computational bottleneck is the algorithm to calculate node re-ordering, edge removal, or the dynamic process, that are done in a pre-processing stage. Therefore, computational scalability depends more on the specific choice of algorithms than on the visualization stage. For example, the recurrent neighbors strategy is faster for small-to-medium size networks than the optimised MSV because the first is deterministic and the second runs over several configurations (simulated annealing) to find the optimal solution (ELZEN et al., 2014). For larger networks (with thousands of nodes), both methods require intensive calculations and naive solutions (e.g. appearance or lexicographic) may work relatively faster. Such computational limitations may also hamper the applicability of such methods on online visualization of real-time data.

5.3 Final considerations

Network visualization provides qualitative visual insights about the network structure to support identifying non-trivial connectivity patterns and developing new statistical measures. The visualization of dynamic processes unfolding on the network also helps

to trace trajectories, transmission paths and the evolution of the states of nodes. The visualization of temporal networks are particularly helpful since patterns of node and edge activity are typically highly irregular in time. Several visual layouts have been proposed to view temporal networks but all have limitations due to their multiple degrees of freedom. The main challenge is that visual information scales with network size and edge density causing visual clutter due to edge overlap, node proximity and fine temporal resolution.

In this chapter we explored the TAM layout for visualization of dynamic process, a strategy that consists on the complete removal of edges and visualization of node properties (e.g. node activity, structure or states) using colors and gradients on nodes. TAM is useful to identify activity patterns such as temporal clusters of activity and periods of inactivity and was also used to simultaneously visualize temporal networks and dynamic processes. This method revealed non-trivial random walker trajectories (e.g. being trapped between two nodes over time), allowed mapping of infection transmission paths and the identification of timings and directions of infection events. The temporal layouts discussed in this chapter are naturally limited to a few thousand nodes, hence alternative scalable strategies are necessary to handle big network data.

Nevertheless, these solutions have a range of applications on mid-size network data such as those frequently used in social systems (friendship, proximity contacts, opinion dynamics), economics (inter-bank loan networks, transportation, cascade failures), business (enterprise partnerships, corporate board directors), public health (contact tracing, impact of vaccination/intervention, epidemics), or biology (neuronal activity, signaling), to name a few possibilities.

Chapter 6

User experiment for evaluation of temporal network visualization techniques

This chapter introduces a user study, where the goal is to summarize the most known representations for temporal networks and to find the relation of the layouts with the most appropriate tasks.

6.1 Design of user study

Several studies over the years have focused on proposing novel visualization techniques for analyzing patterns, trends, and anomalies in complex networks. However, only few of them conducted experiments in which a user evaluates the visualization proposals for static (GHONIEM; FEKETE; CASTAGLIOLA, 2004; GHONIEM; FEKETE; CASTAGLIOLA, 2005; KELLER; ECKERT; CLARKSON, 2006) and temporal (COL et al., 2018; BOYANDIN; BERTINI; LALANNE, 2012; KERRACHER et al., 2015) networks. The goal of these user studies is to find some particularities for each layout. For instance, Ghoniem et al. (GHONIEM; FEKETE; CASTAGLIOLA, 2004) compares a node-link diagram with a matrix-based representation and concludes that for graphs with more than twenty vertices, matrix representations are better than a node-link diagram, however, to find paths in networks, node-links are preferable by the users. Other studies created taxonomies to identify the best visualization techniques according to given datasets and tasks (BECK; BURCH; DIEHL, 2013; BACH; PIETRIGA; FEKETE, 2014; AHN; PLAISANT; SHNEIDERMAN, 2014). The authors (BECK; BURCH; DIEHL, 2013) first introduced some criteria to analyze temporal networks, categorizing into three dimensions: general, dynamic and scalability. Furthermore, the authors (BACH; PIETRIGA; FEKETE, 2014) presented a storyboard (small multiples) with several design choices for temporal representations, introducing an alternative taxonomy dividing tasks into tem-

poral tasks (when), topological tasks (where), and behavioral tasks (what).

Each layout, method or technique for temporal networks has a good use for a specific task, however, identifying which one is ideal for a given task is far from being trivial (BECK et al., 2014). A proper way to identify this relation is to conduct a user study, where, according to a given task, the user is capable of choosing what is the best representation in a context. The authors (BECK et al., 2014) divided the layouts into two main categories to evaluate temporal networks: Animation and Timeline. Animation refers to a sequence of pictures that are set in order to appear as moving images (animation over time) and Timeline is a line representation that shows in a static image (an image containing all the timestamps).

We propose a user experiment to evaluate visualizations techniques for temporal networks using animation and timeline layouts. To represent the animation we chose Structural (Orman et al., 2014) and Matrix (BEHRISCH et al., 2016) and to timeline Massive Sequence View (MSV) (ELZEN et al., 2014) and Temporal Activity Map (TAM) (LINHARES et al., 2017b), that were already used in several research areas (LINHARES et al., 2019b; LINHARES et al., 2017b; GHONIEM; FEKETE; CASTAGLIOLA, 2004; BEHRISCH et al., 2016; LIN et al., 2008; AHN; BAGROW; LEHMANN, 2010) (more details about the layouts in Sec. 2.1.1). Our study also uses the taxonomy (AHN; PLAISANT; SHNEIDERMAN, 2014) to ensure that the tasks were diversified. To the best of our knowledge, our study is the first to combine all those criteria. This study tries to find a relation between these layouts with specific tasks, identifying the most appropriate tasks and the user's preferences. The idea is to present some tasks and the user must evaluate through a questionnaire how suitable a particular layout is for that task.

We run the experiment with undergraduate and postgraduate students from different backgrounds. First, an instructional video was recorded to ensure that every user could have access to the same concepts, not depending on previous knowledge. The addressed concepts in the video are the definition of graphs, temporal networks, temporal layouts (the four chosen layouts, details in Sec. 2.1.1), with basic examples applied in real networks teaching how a user can interpret data for each layout. Furthermore, the video contextualizes the two used networks (Primary School and Hospital) and presents a brief tutorial on how to answer the online questionnaire. The video length was 6min 44s. For standardization, the participants were asked to follow only the instructions of the video without consulting any other source.

The user answered a questionnaire with four questions for two different networks. For each network, the user was capable of navigating into four different layouts (MSV, TAM, Structural animation, Matrix animation). For each question and each layout, the user rated the layout using a Likert scale to interpret the task described in the question (see Tab. 12). In addition to the collected answers, the user was required to set a trust

rating for each question and layout using a 5 Likert Scale (Tab. 12). We also collect extra information, such as age, gender, and previous knowledge in the given areas of the study. In the end, we asked the participants to rank the four layouts in three different categories: (i) layouts more likely to lose information; (ii) preferred layout; and (iii) easy to use. Optionally, users could give feedback about the entire process. We decided to not use time constraints to conduct the experiment, as we expect users from different backgrounds and knowledge in the study field.

Table 12 – Likert scale for network and trust rating questions.

Likert scale	**1**	**2**	**3**	**4**	**5**
Network questions	Very difficult	Difficult	Neutral	Easy	Very easy
Trust rating questions	Very low	Low	Neutral	High	Very high

Due to the complexity of the study, which involved four questions and four layouts for each of the two networks, some design choices were made to avoid a long time duration of the experiment, which could result in a lack of interests by the participants. We decided to limit the interaction tools, restricting then only to zoom and pan for the static images, and animation control, such as moving forward and backward in the time, for the animations.

6.1.1 Layouts decisions

The construction of the four layouts were based on methods, techniques and parameters used in previous studies to enhance the visualization in these layouts. Also, whenever possible, it was used the same parameters in all layouts, such as node organization and the same color to represent the node's metadata. Node ordering is a very important technique to enhance layouts of MSV, TAM and Matrix animation (BEHRISCH et al., 2016; LINHARES et al., 2019b). Similarly, the Structural animation requires a node organization in the two-dimension space to enhance the analysis (BATTISTA et al., 1994). So, in order to organize the nodes in all layouts, we focused on community-based algorithms, such as CNO (Chapter 4) for MSV and TAM. In animated layouts, to keep the mental map the animations were built considering the transition speed and the fading factor (BECK et al., 2016; BECK et al., 2014). Figures 49(a-b) illustrate the fading factor for the Structural (Fig. 49(a)) and Matrix (Fig. 49(b)) layouts. At each time, nodes, edges, and matrix cells are highlighted according to the activated time. For example, cells A, B and B, A are active in the first time step. At the second time step, a new connection is formed (A, D and D, A) and a fading factor, i.e., a transparency effect, is applied to connections from previous timestamps. Without this effect, nodes and edges would be just blinking at each time step on the screen, and would be harder to maintain the mental map.

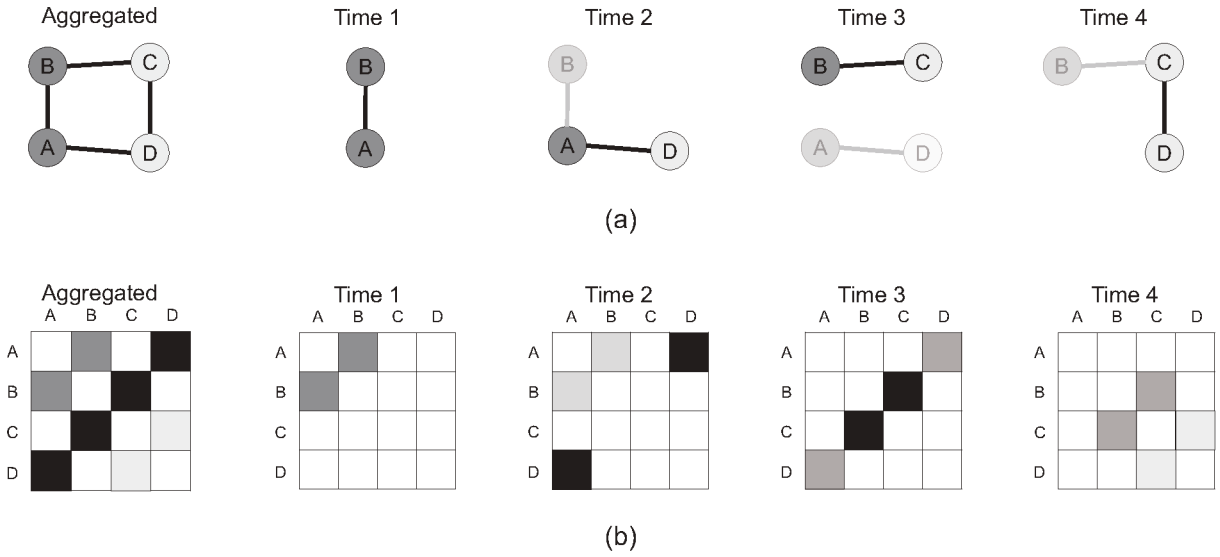

Figure 49 – Animation illustrating the fading factor effect for each timestamp. (a) Structural animation and (b) Matrix animation.

6.1.2 Questionnaire

The guideline for preparing the questionnaire was a taxonomy proposed for network evolution analysis (AHN; PLAISANT; SHNEIDERMAN, 2014). The authors created a three dimensions taxonomy and their respective subcategories to represent different tasks, such as entity, property, and temporal features. We use this taxonomy focused on the temporal features. This taxonomy was chosen to diversify our questions and to ensure different types of tasks for temporal networks. The three main categories (and four subcategories) that form this taxonomy were used:

- **Individual Events - Birth and Death:** Represents a case of individual events identifying the beginning and the end of an event. It can be used to find when groups or a specific entity appear or disappear.

- **Shape of Changes - Peak or Valley:** Represents a case of the shape of changes, tracking movements of sudden increase or a decrease. It can be used to compare or identify peak and valley events.

- **Shape of Changes - Repetition:** Represents a case of the shape of changes, tracking the repetition of patterns along time. It can be used to compare or identify this repetition patterns.

- **Rate of Changes - Speed:** Represents a case of the rate of changes, defining if the speed of an entity is a fast or a slow event. It can be used to compare the speed between different entities, finding which one was faster or slower.

Table 13 – Questionnaire for school and hospital networks and their taxonomy categorization.

Questionnaire - Primary School Network	**Taxonomy**	**Subcategory**
SQ1) One day at school has one period of activity followed by inactivity because this school doesn't have classes at night. The school network has 2 days. How do you evaluate the task of noticing that the network has two days?	Shape of Changes	Peak/Valley
SQ2) The school has activity in the mornings, afternoons and there is lunch time. Lunch time is characterized by fast and high information exchange between different classes. How do you evaluate the task of detecting the lunch time in school?	Rate of Changes	Speed
SQ3) Classes that leave early in school are represented by the continuous absence of activity on the same day. Two classes leave early on the second day of school. How do you evaluate the task of identifying them?	Individual Events	Birth/Death
SQ4) Teachers actively communicate during the day with students and even with other teachers. How do you evaluate the task of following the communication involving the teachers (nodes in black)?	Individual Events	Birth/Death
Questionnaire - Hospital Network	**Taxonomy**	**Subcategory**
HQ1) One day at the hospital has a period of high activity (daytime period) followed by low activity (night-time period). How do you evaluate the task of noticing that the network has 5 days?	Shape of Changes	Peak/Valley
HQ2) The hospital staff has different starting and finishing work times. How do you evaluate the task of following the starting and finishing times of the medical staff?	Individual Events	Birth/Death
HQ3) Different patients and nurses communicate between themselves along all days in the hospital. How do you evaluate the task of identifying the several interactions between patients and nurses?	Shape of Changes	Repetition
HQ4) A new day in the network is identified by the abrupt change from the absence of activity to periods with activity. How do you evaluate the task of noticing that it is beginning a new day in the hospital?	Rate of Changes	Speed

A total of 8 questions were elaborated (four questions per network). Three for Individual Events, three for Shape of Changes and two for Rate of Changes, as detailed in Tab. 13. The questions were organized from simple to harder questions. The idea is that the user could learn how to use and interpret the layout in simpler tasks and then understand more easily the harder ones. The question descriptions were carefully designed to guide the user to observe the right task, first explaining *what* was the pattern and then asked to evaluate that task in the four layouts.

6.1.3 Web system

Besides the instructional video, the users accessed two online web page containing four layouts for each network (see Fig. 50) and a questionnaire (see Fig. 51). They were allowed to navigate between the layouts and answers. The experiments were conducted using computers with large monitors (at least 24 inches) from the laboratories of the Faculty of Computing, Federal University of Uberlândia, Brazil.

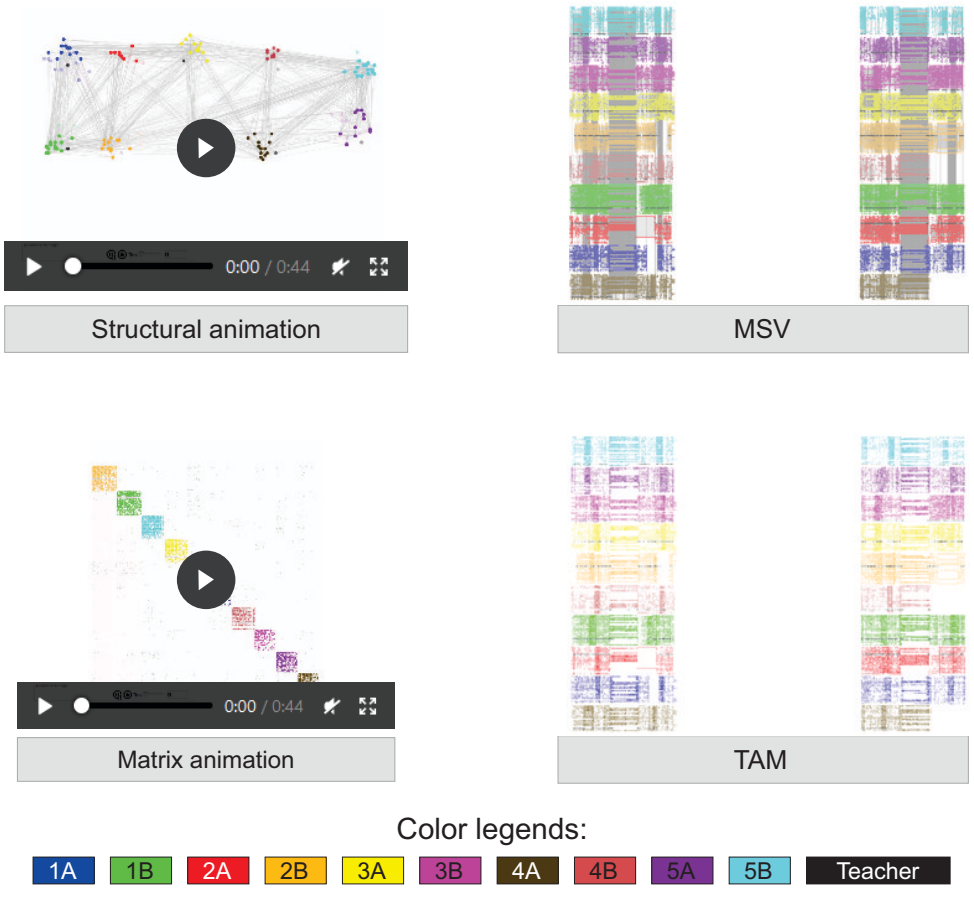

Figure 50 – Example of the four layouts used in the user evaluation.

Network - Question #

*Question description:

	Very difficult	Difficult	Neutral	Easy	Very easy
Structural animation	○	○	○	○	○
MSV	○	○	○	○	○
Matrix animation	○	○	○	○	○
TAM	○	○	○	○	○

*Trust rating:

	Very low	Low	Neutral	High	Very high
Structural animation	○	○	○	○	○
MSV	○	○	○	○	○
Matrix animation	○	○	○	○	○
TAM	○	○	○	○	○

Figure 51 – Example of the questionnaire used in the user evaluation.

6.1.4 Participants

Before applying the questionnaire, we sent the evaluation to five expert users, for an initial evaluation about the answer time, the environment to execute the experiment, and the usability of our web system. We received several feedbacks from the experts related to improvements of the question's description, web system design, video length, among others. All these suggestions were included in the final version of the experiment.

A total of 74 students participated in the experiment. They are from different backgrounds – such as computer science, business, electrical engineering – and from different levels, undergraduate or graduate courses. All of them were students from Federal University of Uberlândia, Brazil. The experimental procedures were performed according to the Ethics Committee of the Federal University of Uberlândia, under the Certificate of Presentation for Ethical Consideration (CAAE – number 98260818.1.0000.5152).

6.2 Results

This section describes the statistical analysis of the Likert scale and discuss about the user experience feedback.

6.2.1 Experiment

The time spent by the users to complete the experiment is shown in Fig. 52(a). This time includes only the time in which the users interacts with the questionnaires. The video length was not included in the total time as the user was allowed to see the video more than once for better understanding. On average users spent 19.72 minutes to answer all

questions. We also evaluated the time spent for each questions (Fig. 52(b)). The questions (SQ1 – SQ4) were related to the school network and (HQ1 – HQ4) were related to the hospital network. Although the average time to answer each question was 2.11 minutes, there is a clear decrease in the time spent after each question. This is expected since, as we maintain the same layouts for the same networks, after each question the user is already acquainted with the layouts and with the visualization techniques. Notice that, in question HQ1, the first question of the Hospital network, it is presented to the user four new layouts, increasing again the time spent to answer the questions, but not as high as the first question (SQ1), because the user is already familiar with the visualization techniques.

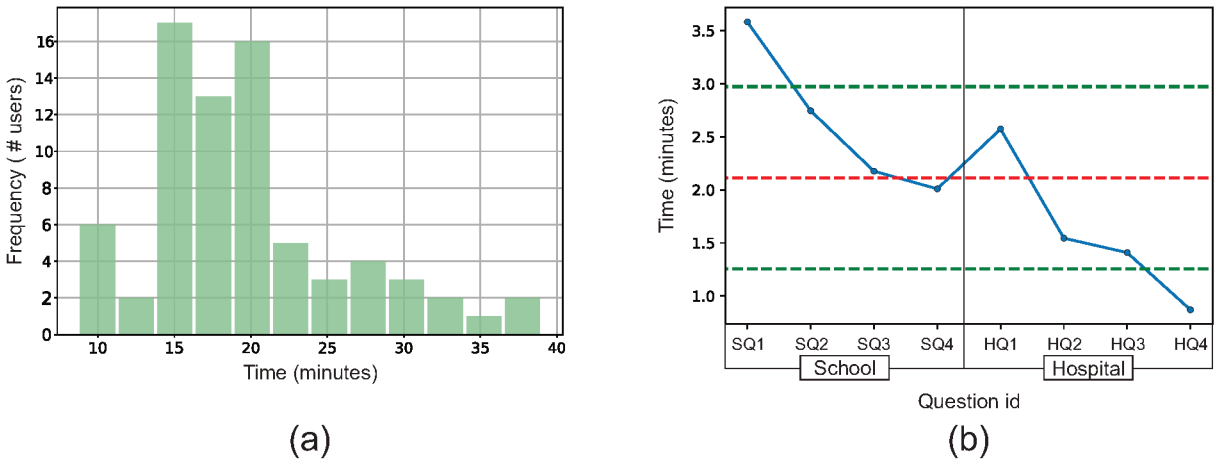

Figure 52 – Time spent by users to execute the experiment in relation with: (a) number of users; (b) question identification. In (b), red lines represent the mean and the green lines represent the standard deviation.

In order to show the participants answers, we use the diverging stacked bar charts, as it represents an appropriated visualization for Likert scale data (HARPE, 2015). Figure 53 shows the answers for the school network in terms of the percentage of the 5-Likert scale answers for each question (the answers for who found easy to analyze the layout are shown in the right of the zero using positive values, while who found difficult is shown to the left using negative values). For the first (SQ1) and third (SQ3) questions, every layout was classified as Easy/Very easy to complete the tasks for the majority of the users. However, we found divergent answers for the second (SQ2) and fourth (SQ4) questions. The TAM layout for SQ2 presented by far the worst result. This is justified since the question is related to the identification of the lunch time, which is characterized by the intense interaction between nodes, a feature represented by edges, which is not presented in TAM layout (recall that TAM layout hides all the edges in the layout to emphasize node activity). For the same question, SQ2, MSV and Matrix animation presented inconclusive results, and Structural animation was by far the best result. In the fourth question (SQ4),

Matrix animation presented the worst result, with Structural animation also presenting a small advantage with respect to the other layouts, although the result was not so strong as in the second question (SQ2).

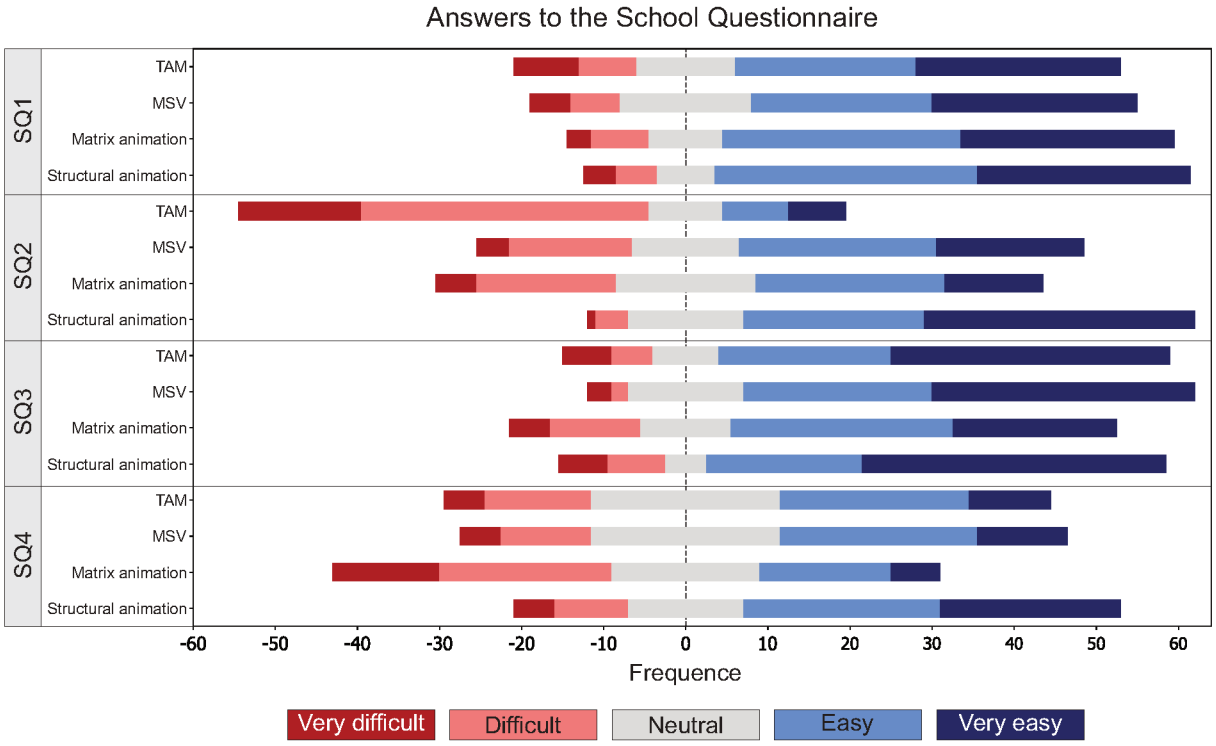

Figure 53 – Participants answers in school network using a diverged stacked bar.

Figure 54 represents all the answers for the hospital network. Differing from the school network, where the Structural animation was considered easy in almost every question, on the hospital network different layouts presented different results. In the first (HQ1) and third (HQ3) questions, TAM and MSV presented better results when comparing with the animations. In the second question (HQ2), no significative differences were found between the layouts, with a tendency for easy answers. In the third question (HQ3), except for the Structural animation, all the other layouts presented the worst result of this network.

In addition to theses analyses, we evaluated the answers using median and interquartile range (IQR) statistics (CLARK-CARTER, 2005). For this computations, the Likert scale was converted to numerical values, according to Tab. 14. The median refers to the middle value of the answers, and the IQR refers to dispersion of the data, i.e., how far the numbers are from the normal distribution, which varies from 0 to 5 in a 5-Likert scale. High IQR values indicates a high dispersion, which can represent a disagreement between the participants indicating that the median result is not significant.

Table 15 shows the median and IQR for answers of the School network. The only layout to be considered very easy with very low IQR was the Structural animation in the third question (SQ3, Mdn=5, IQR=1.0). This means that this layout is ideal to represent

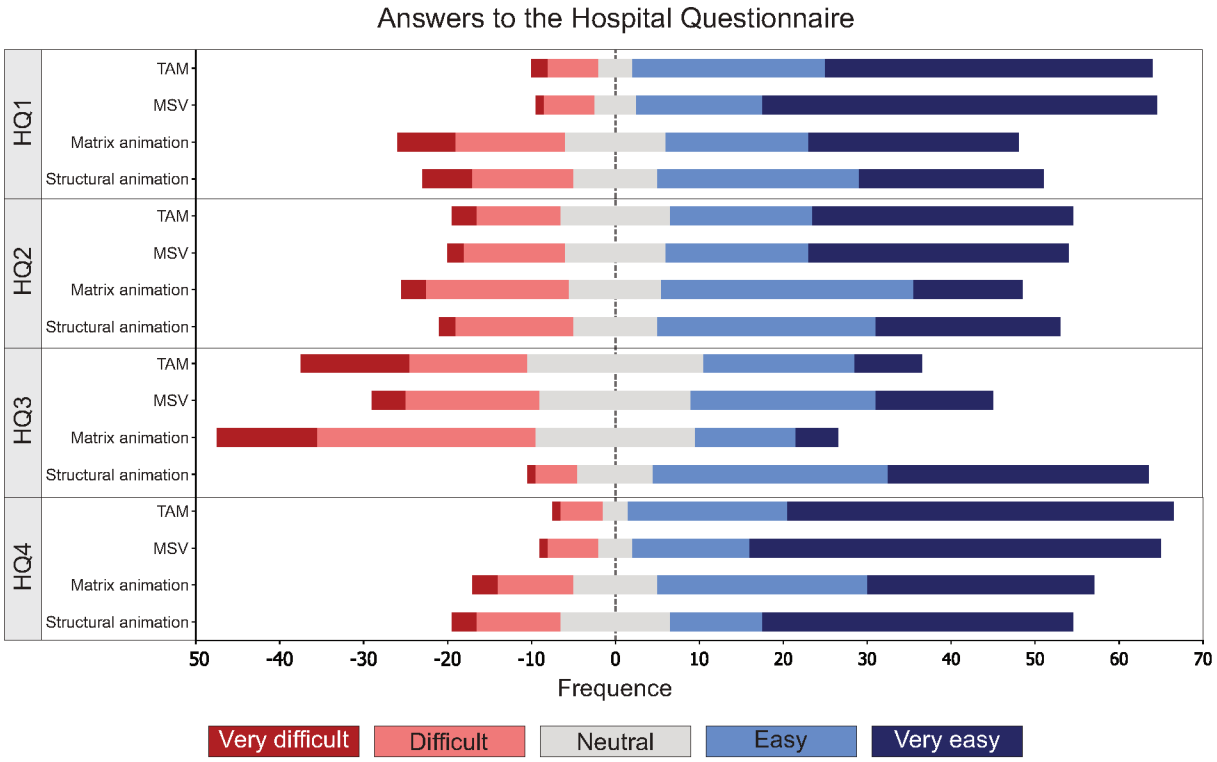

Figure 54 – Participants answers in hospital network using a diverged stacked bar.

Table 14 – Likert scale in numerical values.

Value	**Likert scale**
1	Very difficult
2	Difficult
3	Neutral
4	Easy
5	Very Easy

that task for this network. Also, the only result above the neutral (easy and very easy) with low IQR was the TAM, presenting a difficult layout to represent that task. Although all layouts were considered easy to perform the task in the first question (SQ1), Structural animation also presented the lowest IQR. In the fourth question (SQ4), all layouts were considered neutral, except the Structural, which was considered easy with a 2.0 of IQR and can represent some variance between the users.

Table 16 shows the median and IQR for answers of the Hospital network. In the first (HQ1) and fourth (HQ4) questions, when analysing the median, all layouts were considered easy or very easy, however, the Matrix animation presented the highest IQR (3.0) for the first question (HQ1), which can indicate a high disagreement among the users. In the second question (HQ2) none of the layouts stood out, i.e., all results were the same (Mdn=4, IQR=2.0). In the third question (HQ3), the Matrix animation was categorized as difficult by the median with a low IQR, following the diverging stacked bar

Table 15 – Median (Mdn) and interquartile range (IQR) for the answers of School network.

	SQ1		SQ2		SQ3		SQ4	
	Mdn	**IQR**	**Mdn**	**IQR**	**Mdn**	**IQR**	**Mdn**	**IQR**
Structural animation	4	1.0	4	1.75	**5**	**1.0**	4	2.0
MSV	4	2.0	4	1.75	4	1.75	3	1.0
Matrix animation	4	1.75	3	2.0	4	2.0	3	2.0
TAM	4	2.0	**2**	**1.0**	4	1.75	3	1.0

charts results in Fig. 54.

Table 16 – Median (Mdn) and interquartile range (IQR) for the answers of Hospital network.

	HQ1		HQ2		HQ3		HQ4	
	Mdn	**IQR**	**Mdn**	**IQR**	**Mdn**	**IQR**	**Mdn**	**IQR**
Structural animation	4	2.0	4	2.0	4	1.0	5	2.0
MSV	**5**	**1.0**	4	2.0	3	2.0	**5**	**1.0**
Matrix animation	4	3.0	4	2.0	**2**	**1.0**	4	2.0
TAM	**5**	**1.0**	4	2.0	3	2.0	**5**	**1.0**

6.2.2 Users feedback

At the end of the evaluation, the users ranked each layout according to different criteria. We propose a score to evaluate these rankings, giving weight to the participant's responses. For example, if a layout is ranked in the first place, we give this layout a score of four. Second, third and fourth places are scored with three, two and one respectively. The final result is a value between 1 and 0, which represents a sum of the participant rankings normalized by the highest score value.

Figure 55 shows the score to three following criteria: easy to use, preference and prone to lose information. The results from easy to use and preference were almost the same, indicating that the participants evaluated those two criteria in the same way. To these two criteria, higher values indicate better results. The Structural animation presented the best results, followed by MSV. Matrix and TAM layouts presented the worst results according to the participants. In the case of the layout prone to lose information (Fig. 55(c)), lower values indicate better results. The layout most voted to lose information was the Matrix, followed by TAM. In this case, MSV layout was considered the best layout, closely followed by the Structural animation.

Also, users could leave general comments or suggestions regarding the study. In general, most comments were related to positive or negative points for each layout (Tab. 17). Several aspects were raised by the users, such as layouts that are polluted, confuse, inter-

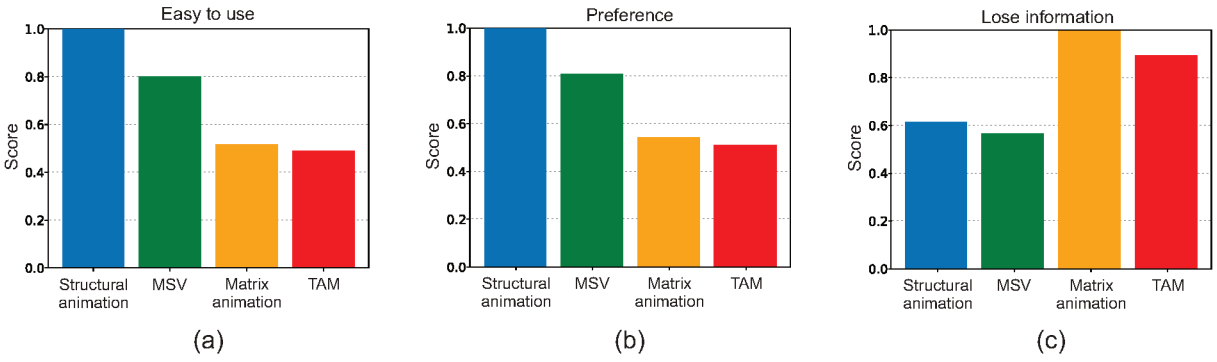

Figure 55 – Layouts ranked by the users according to: (a) Easy to use; (b) Preference; (c) Prone to lose information.

esting, simple, and so on. Three of the four layouts had 5 positive points, and only the Matrix had one positive point. Also, the Structural animation had the fewest negative points, similar to the results of Fig. 55.

Table 17 – Users feedback highlighting positive and negative points for each layout.

Layout	Positive points	Negative points
Structural animation	Easy Good to analyze time Easy to interpret Good to see interactions Easy to predict pattern changes	Lose information
MSV	Easy to compare information Good to analyze time Interesting Don't lose information Clear in the interactions	Polluted Hard to read the communication among nodes Less interesting than Structural animation Complex
Matrix animation	Good	Complex Bad with low communication among nodes Polluted Messy Hard to understand
TAM	Simple Good to compare information Good to analyze time Don't lose information Better than Matrix animation	Difficult to read the communication among nodes Complex Confuse Strange

6.3 Discussion

Based on the results of this chapter, Tab. 18 summarizes the best and worst results of the layouts in comparison with the taxonomy. There were two layouts not recommended in specific cases, the first was the Matrix animation to the Repetition task, and the TAM to the SQ2 question, although when comparing in the same taxonomy, to the Hospital network this layout was recommended. A similar case occurs to the Peak/Valley category, where for School network (SQ1) the animated layouts were preferred, while

for Hospital network (HQ1) the temporal layouts were the best. In these cases, the characteristics of the networks, such as the size of nodes, distribution of edges, among others, appear to be more important than the taxonomy to determine the best layout. In the Birth/Death category, the Structural animation was preferred in all cases, although for Hospital network (HQ2) all layouts presented good results. Similarly, MSV was the winner of the Speed category. In general, Structural animation, MSV and TAM presented at least one positive case for all three main categories, with only Matrix animation not presenting a positive result for the Rate of Changes. Furthermore, the Individual Events category only presented recommended cases.

Table 18 – Relation between layouts and taxonomy.

	Shape of Changes			**Rate of Changes**		**Individual Events**		
	Peak/Valley		**Repetition**	**Speed**		**Birth/Death**		
	SQ1	**HQ1**	**HQ3**	**SQ2**	**HQ4**	**SQ3**	**SQ4**	**HQ2**
Structural animation	✓		✓	✓		✓	✓	✓
MSV		✓		✓	✓			✓
Matrix animation	✓		✗					✓
TAM		✓		✗	✓			✓

6.4 Final considerations

The objective of this user evaluation is to provide insights into the relation between layouts and tasks. Although there is no clear relation yet, we found some interesting results. In general, the layouts were classified more as very easy than very difficult, indicating that all layouts can be good to represent temporal tasks. Also, for some specific tasks, some layouts stood out negatively and others positively, indicating a very bad or a very good layout in comparison with the others for that task. Also, according to the users' feedback, the Structural animation was the best considering some aspects, such as easy to use, and preference, followed by the MSV. Matrix and TAM have worst results in some cases.

Chapter 7

DyNetVis System

This chapter describes the *Dynamic Network Visualization (DyNetVis)*[1], which is an interactive software for temporal network visualization. It implements different layouts, by using state-of-the-art techniques to enhance visualization and improve data interpretation. DyNetVis also implements several techniques to enhance these layouts, and dynamic processes, including infection models to analyze spreading of opinion, rumors, diseases, and others.

7.1 Software description

This section presents requirements for using DyNetVis and explains its workflow. Then, we present the available layouts and the state-of-the-art techniques and interaction functionalities that allow the visual analyses.

7.1.1 About the software

DyNetVis was developed in Java programming language, aiming the support for different platforms. The development environment used was Netbeans. The software was developed using an open-source graph library called jgraphx[2], that provides functionalities for graph visualization and interaction using Java. The file format[3] required to open a network is three tab-separated columns, containing the two nodes and the timestamp, where each line represents an edge in a specific timestamp.

Users are expected to have previous knowledge about temporal networks. Domain experts with knowledge about the network data can better understand the meaning of specific patterns. Users with less knowledge about the data usually tend to identify some general properties, such as global behaviors or relations among different timestamps.

[1] www.dynetvis.com

[2] https://github.com/jgraph/jgraphx

[3] More details about files format can be found in the documentation: <https://github.com/claudiodgl/DyNetVis/blob/master/readme.pdf>.

Dynamic data visualization and interaction provide the capability of interpreting and understanding data through exploratory analyses, by choosing techniques that better fit for a specific task. In this way, it is possible to identify non-trivial patterns, trends, and anomalies in the network structure, which could be more challenging without appropriate visualization techniques. As visualization can be used in several disciplines and networks can be modeled in different contexts, DyNetVis was used for exploring networks in a variety of research areas, such as healthcare (LINHARES et al., 2017a; LINHARES et al., 2019b), a corporate bankruptcy (LINHARES et al., 2017b), and social networks (LINHARES et al., 2019b; LINHARES et al., 2017b). DyNetVis was also used to visualize dynamic processes, allowing the user to understand the impact of the visualization in the studied network (LINHARES et al., 2019a).

There are other software to visualize complex networks, for example, Gephi (BASTIAN; HEYMANN; JACOMY, 2009), Cytoscape (SHANNON et al., 2003), and libraries for programming languages R and Python. Some of them include techniques for temporal networks, such as DyNet Viewer (LI et al., 2017), a Cytoscape plugin that only provides the structural layout, or Gephi, that provides the structural layout with animation. Other proposals, such as GraphChi (KYROLA; BLELLOCH; GUESTRIN, 2012) and Pregel (MALEWICZ et al., 2010) are specialized in computations for large networks but without focusing on visualization. In fact, none of the proposed software are focused on layouts with temporal information nor provide visualization techniques that facilitate the analysis of temporal networks.

7.1.2 Visual analysis workflow

The visual analysis workflow is illustrated in Fig. 56. First, the user selects a layout and then load the network by opening a file or by generating a new random network. Next, the user optionally chooses between time-related filters, such as initial and final time, and the temporal resolution. Then, the selected layout is constructed and exhibited, and the user can freely navigate and explore the network by choosing specific techniques to reduce visual clutter or to highlight parts of the network structure, such as node ordering, edge filtering, and dynamic processes. Interaction tools, such as the selection of specific nodes, edges, and time, are used to highlight regions of interest (more details of techniques and interaction tools in Sec. 7.1.3.1). The user can study the produced layouts using statistical or visual analysis and compare combinations of layouts and techniques through exploratory analyses. Furthermore, the user can export to an image the layout under analysis, which can be used for future analyses.

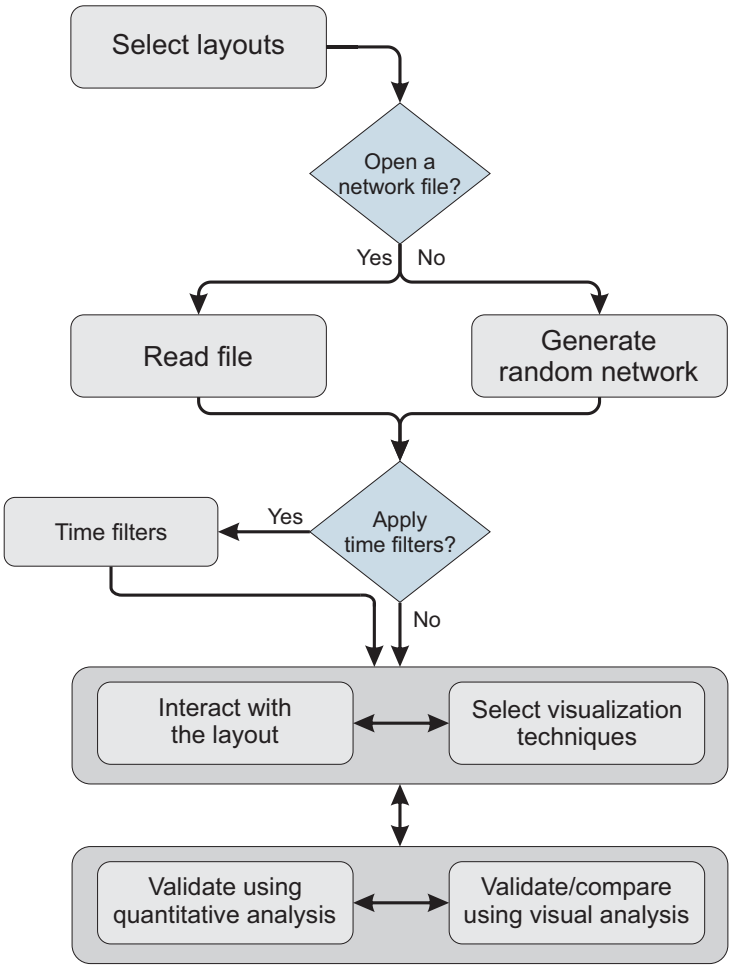

Figure 56 – Representation of DyNetVis workflow diagram.

7.1.3 Visualization techniques

The software implements a set of layouts detailed in Sec. 2.1.1. These layouts can be enhanced using several techniques, that were implemented in the software and exemplified in a sample network (Fig. 57).

7.1.3.1 Software functionalities

DyNetVis implements techniques for enhancing the analysis of the structural, temporal and matrix layouts (Fig. 57). For the structural layout, the software contains techniques to organize the nodes in the layout, such as fast organic, circular and hierarchical (BATTISTA et al., 1994). For the temporal (MSV) layout, DyNetVis offers: node ordering, random, appearance, lexicographic, degree (ELZEN et al., 2013), Recurrent Neighbors (RN) (LINHARES et al., 2017b), and Community-based Node Ordering (CNO) (LINHARES et al., 2019b); the Temporal Activity Map (TAM) heatmap (LINHARES et al., 2017b); edge sampling, such as random, Accept-Reject (AR) (ZHAO et al., 2018), and Edge Overlapping Degree (EOD) (ZHAO et al., 2018). The matrix layout also contains several node ordering techniques, such as trivial, RN and community-based ordering. For

dynamic processes in the temporal layout, DyNetVis includes random walks (STARNINI et al., 2012; ROCHA; MASUDA, 2014) and four epidemic models (SI, SIR, SIS, and SIRS) (BARRAT; BARTHéLEMY; VESPIGNANI, 2008; ROCHA; BLONDEL, 2013).

DyNetVis contains several interactive tools that allow exploring the layouts using zooming and panning or coordinating the analysis between layouts, i.e., simultaneously selecting the same nodes/edges in different layouts. These tools provide different ways to analyse network data from complementary perspectives (leading to different insights). DyNetVis also provides quantitave analysis for visual cluttering, as for example estimation of overlapping edges and edge intersection (LINHARES et al., 2019b), that supports the choice of the most appropriate layouts and ordering algorithms. Table 19 shows a summary of DyNetVis functionalities.

Table 19 – Interactions and functionalities for the three layouts in DyNetVis.

Interaction/Functionality	**Structural**	**Temporal**	**Matrix**
Zoom and Pan	✓	✓	✓
Animation	✓		✓
Coordination between layouts	✓	✓	
Quantitative analysis		✓	
Export image	✓	✓	✓
Depth selection of nodes/edges	✓		
Select specific nodes/edges	✓	✓	
Change shape/size of nodes/edges	✓	✓	
Change color of nodes/edges	✓	✓	✓
Change metadata color of nodes/edges	✓	✓	✓

7.2 Illustrative examples

We illustrate in this section applications using a network that represents interactions between participants of a conference in Turin, Italy, and which contains 113 nodes and 8.892 edges in 3 days (ISELLA et al., 2011). Figure 58(a) shows a screenshot of the temporal layout for the conference network, highlighting the functionalities and interaction tools. Figure 58(b) shows the visual analysis with examples of structural patterns using the temporal layout from Fig. 58(a). A clear division between each day of the network is identified since there are no (or little) activity during the nights (ISELLA et al., 2011). Furthermore, in each day, different patterns are identified depending on the time of the day, such as the transition from inactivity to the increase and decrease of activity in the third day. High activity corresponds, for example, lunch time whereas highly active groups of nodes correspond, for example, to groups of individuals from the same university who

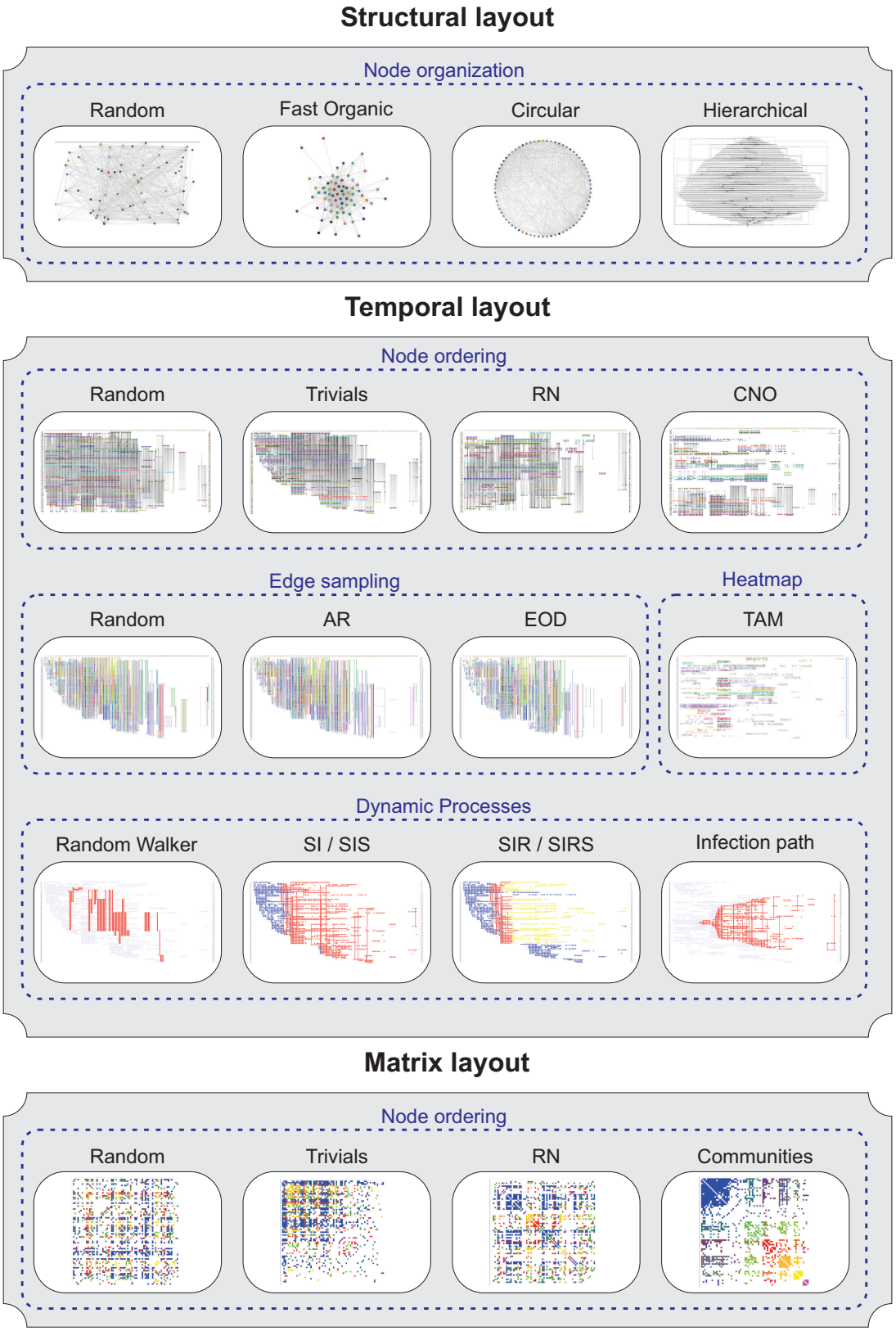

Figure 57 – Techniques implemented in DyNetVis for the structural, temporal and matrix layouts.

are communicating with each other, which can be active in different times, representing, for instance, work shifts among the support staff.

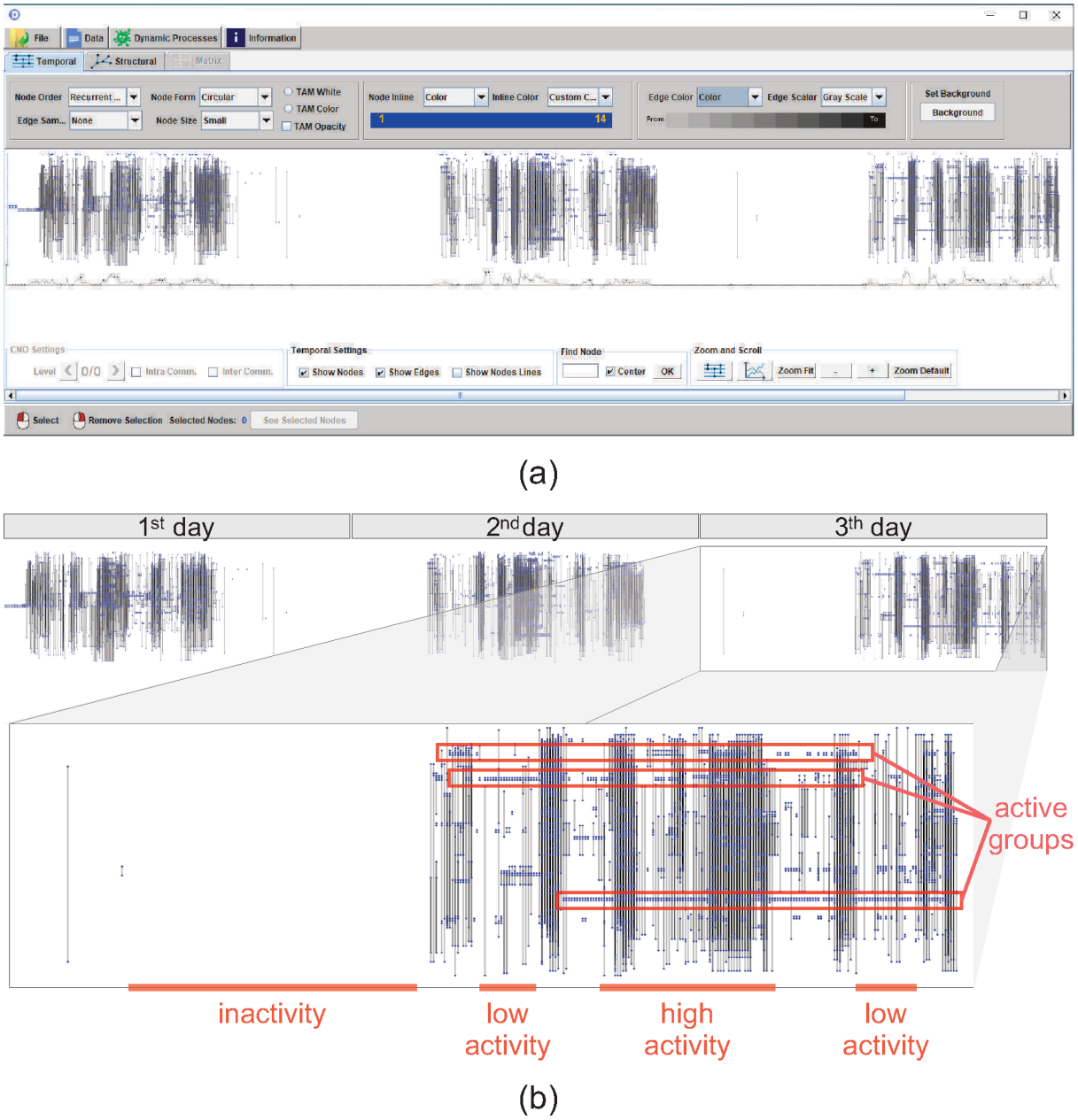

Figure 58 – Application of the temporal layout for the conference network. (a) software screenshot; (b) examples of structural patterns.

Another possibility is to explore dynamic processes in the conference network to study, for example, the spread of fake news or rumors in a network. Figure 59(a) shows the random walker path in three days of the network. In this case, the walker was active in all days, moving within small groups of individuals in the first two days (reflected in the size of the red vertical lines), and hoping between distinct groups in the third day of the network. The nodes closer to each other can be assumed as groups since this is an intrinsic property of the adopted node ordering algorithm, recurrent neighbors (LINHARES et al., 2017b). This pattern can indicate, for example, that groups of individuals were interacting more

often between themselves, but this dynamic changes on the third day of the conference. The SIR epidemic model evolving in the conference network is illustrated in Fig. 59(b). In this case, the infection started on the first day, quickly propagating to almost all nodes in the network. In the second and third days, the infection stopped growing and most of the nodes started to recover from the infection.

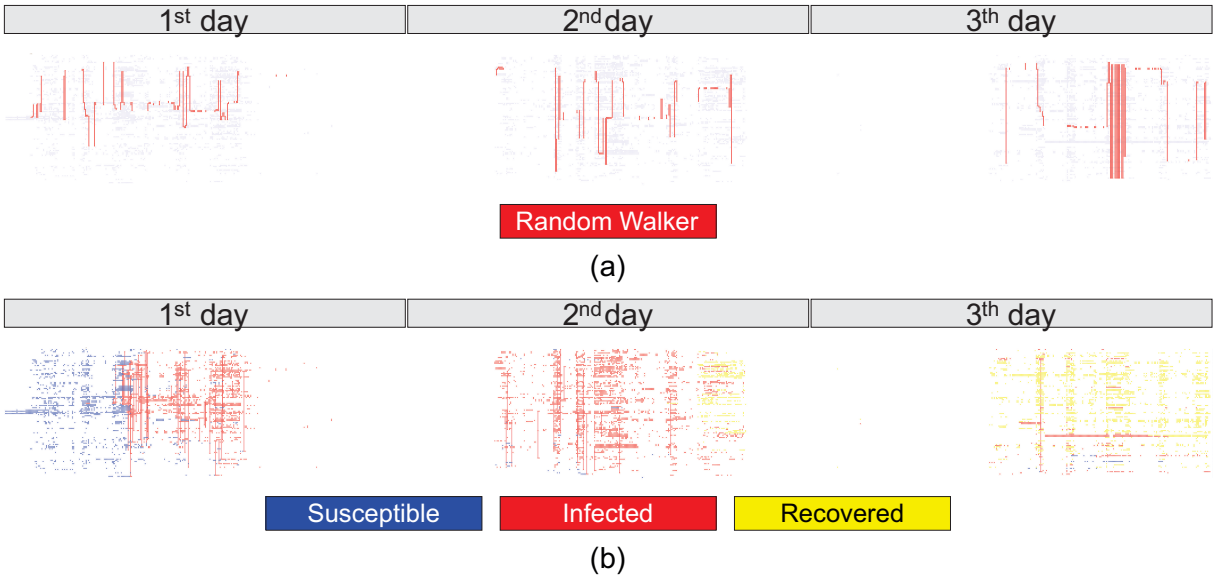

Figure 59 – Examples of dynamic processes in the temporal layout for the conference network. (a) random walker path (in red) and (b) SIR infection model.

7.3 Impact

Data from diverse disciplines can be modeled by temporal networks, as for example, data of relevance for public health, finance, business, informatics, or biology. Data analysis in these disciplines can thus benefit of DyNetVis. For example, in Biology, the visual analysis of temporal networks can be used to study molecular interactions in cells or communication between organisms and species, to understand the behavior of species in a specific environment. In Computer Science, it is possible to model the source code to understand the relation and balance between classes, attributes, and methods. DyNetVis can be used for management and decision making in different contexts. For instance, in a school network, the school principal can observe the interaction between teachers and students or the relationships among students of the same grade/class. In a museum, the manager can observe the moments of high interaction during the day and allocate more support staff for those specific times in the future. DyNetVis can also be used by users without technical knowledge to understand the structure and activity of temporal networks or to teach about networks, visualization, epidemics, and so on (KEIM, 2002).

The applicability of DyNetVis is wide and aids the development of new research questions in every discipline in which it is possible to represent data using temporal networks. The goal of the visualization is to provide insights and help the user in identifying patterns in the data, especially those not previously expected. It is a particularly helpful and insightful tool for qualitative researchers and practitioners not familiar with quantitative and statistical network methods. It is also relevant for educational purposes since visual representations are typically more intuitive for those being introduced to abstract network concepts. Generally speaking, network visualization speeds decision making and facilitates the identification of network patterns. Furthermore, efficient visualization techniques for the temporal layout, such as node ordering and edge sampling, require further studies and new methods, especially related to visual scalability (LINHARES et al., 2019b).

As DyNetVis allows analyses of dynamic processes, structural and temporal patterns can be observed using these techniques. For instance, in cases of modeling the Internet communication route in a computer network, the packet transfer according to the protocol between two devices can be also observed using random walks. Social media can also be modeled into networks and then one can use infection models to simulate social contagions, such as the spread of rumors, information, opinions, or infections. These are just some examples of how a user can use DyNetVis in different contexts, but the field is wide open and more analysis can be elaborated in different scenarios.

DyNetVis allows the integration of visualization techniques to better visualize and explore temporal networks. Furthermore, it can be extended to implement other layouts, such as alluvial (VEHLOW et al., 2015) and circular (ELZEN et al., 2014). All these extensions can also generate new studies and research questions. In cases of large networks, DyNetVis also provides scalable techniques that can help the user to find new patterns, which is also an open field since there are just a few scalable proposals for temporal networks (LINHARES et al., 2019b).

7.4 Human-Computer Interaction analysis

A process of Human-Computer Interaction (HCI) validation was used in DyNetVis, in order to improve the usability and provide better user interaction. This process was created during Human-Computer Interaction graduate classes and was divided into six steps, as shown in Fig. 60. The first step was to conduct a *semi-structured interview* with one student that has a background in networks but was not familiar with the temporal network concepts. After the interview, the student was familiarized with the basic concepts (Section 2.1.1) and was able to participate in the process. In sequence, it was created a *Low Fidelity prototype (Lo-Fi)* and *High Fidelity prototype (Hi-Fi)* of the system interface in order to be evaluated by the student.

The *Lo-Fi process* has detected several mistakes, as features that need to be grouped

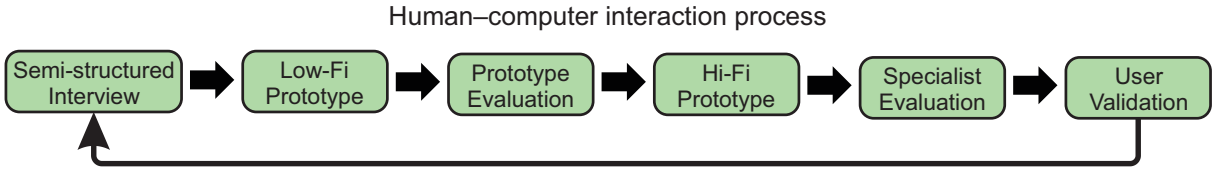

Figure 60 – Human-computer interaction Process for evaluation of the user interface of DyNetVis system.

by similarity, for instance, the show/hide flags of vertices and edges being together, and the need for a better division of functionalities between structural and temporal. The result of *Semi-structured interview* is represented in Fig. 61(a) and Fig. 61(b) represents the result of the *Lo-Fi Prototype evaluation*.

After *Prototype evaluation*, the *Hi-Fi prototype* was implemented based on the best results of the *Lo-Fi process*. Thus, the specialist (the class professor) refined the process, and take into consideration ten very popular usability heuristics (NIELSEN; MOLICH, 1990) in Fig. 61(c) version. Five major problems were detected in the visualization, as for example several issues of break of mental model preservation, problems with the structural and temporal tabs, button positions, and so on. As result, it was created a new version of the *Hi-Fi prototype* (Fig. 61(d)) in a system with intuitive functionalities, improving the user experience.

After the five first steps, it was created an activity among the students of the class, to create a *User validation* of the entire process. Several minor issues were detected with this evaluation, most of then related to the usability of the system and requests for new functionalities to improve the user experience. Fig. 62 shows a comparison between the first version of the software (Fig. 62(a)) and the current version (Fig. 62(b)). It is possible to notice the evolution between the layouts, as for instance the addition of icons and changes in the position of all flags and buttons, to improve the system intuitiveness. In the current version, the system has several features, described in details in Fig. 63 for temporal (MSV) layout, Fig. 64 for structural layout, and Fig. 65 for matrix layout.

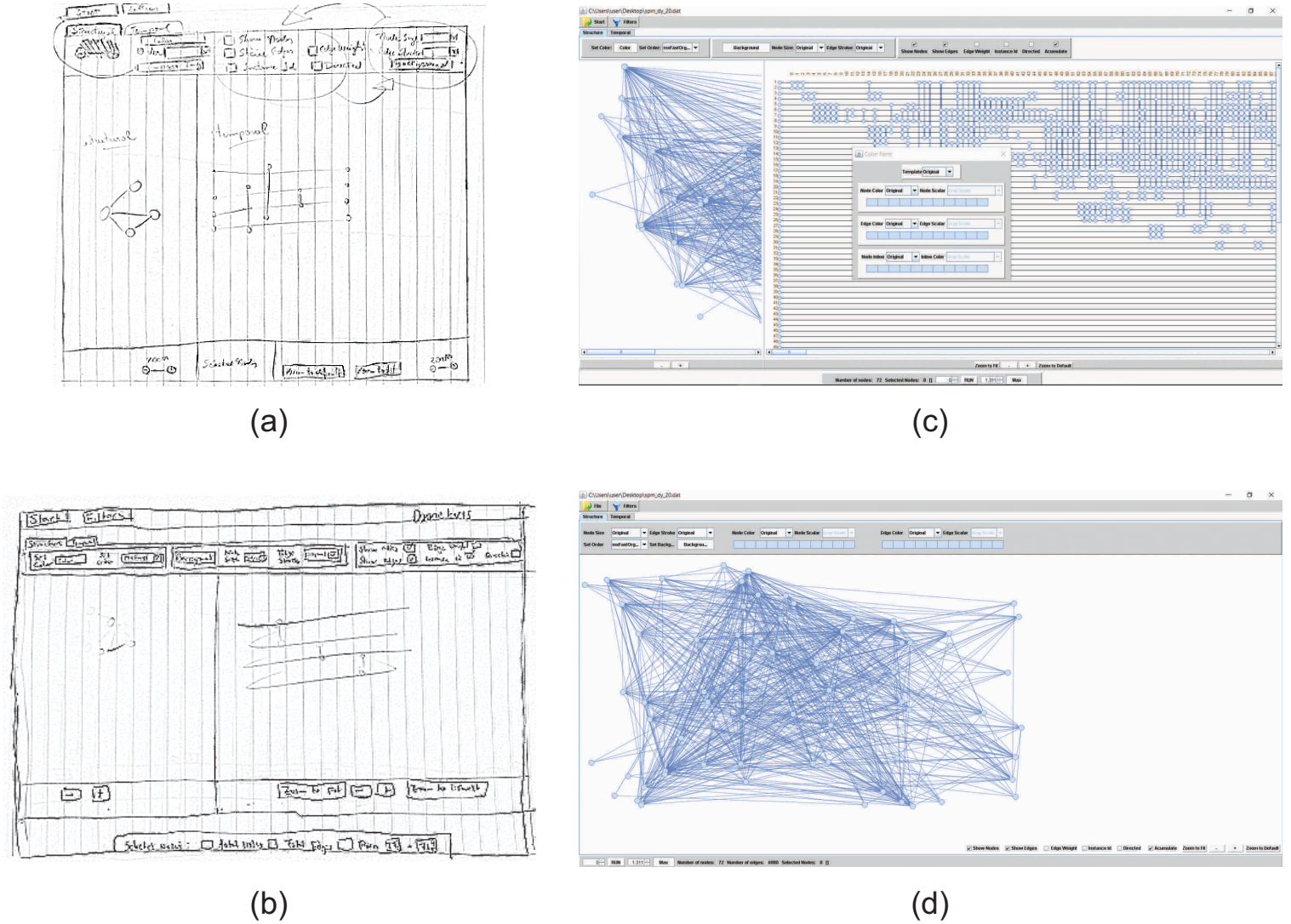

Figure 61 – The second and third steps of the HCI process. The *Lo-Fi prototype*: in (a) is shown the first version and in (b) the second version improved from (a) taking into account the evaluation process. *Hi-Fi prototype*: in (c) is shown the first version and in (d) the second version improved from (c) after *specialist evaluation*.

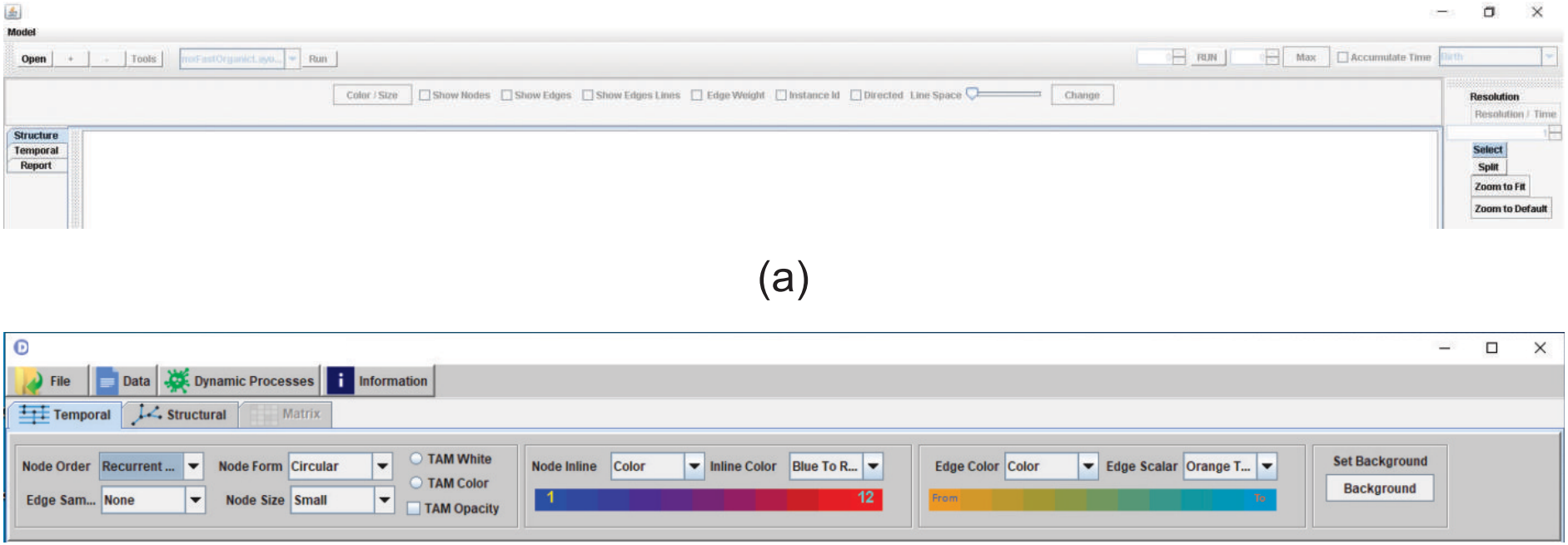

Figure 62 – DyNetVis layout comparison between the first and the current version analyzed. In (a) the first version of the DyNetVis system without the HCI study; in (b) the final version after the six steps of the HCI. Several notable changes can be perceived, as for instance the addition of icons to improve the intuitiveness of the software.

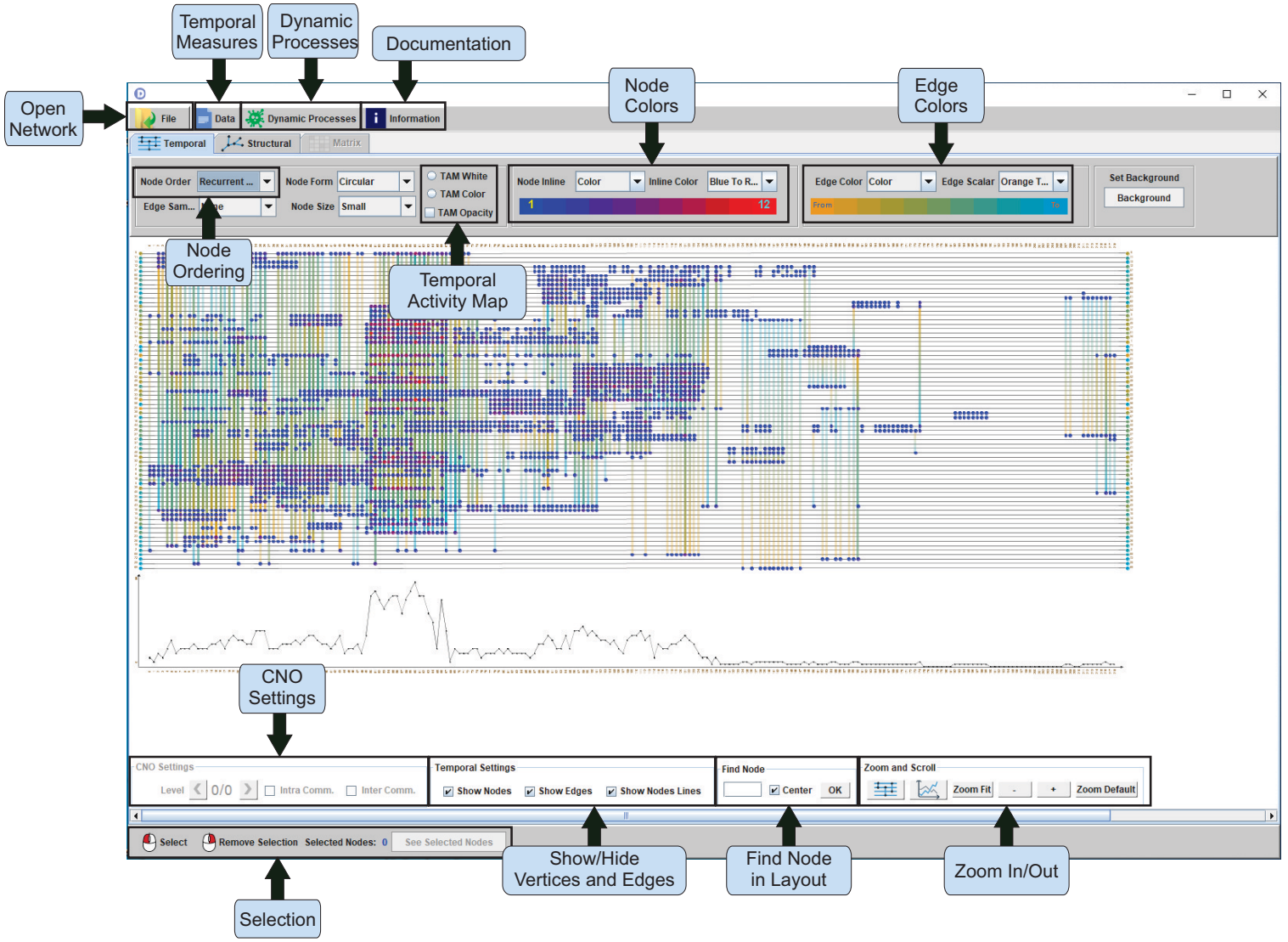

Figure 63 – Temporal (MSV) layout of DyNetVis demonstrating the system features.

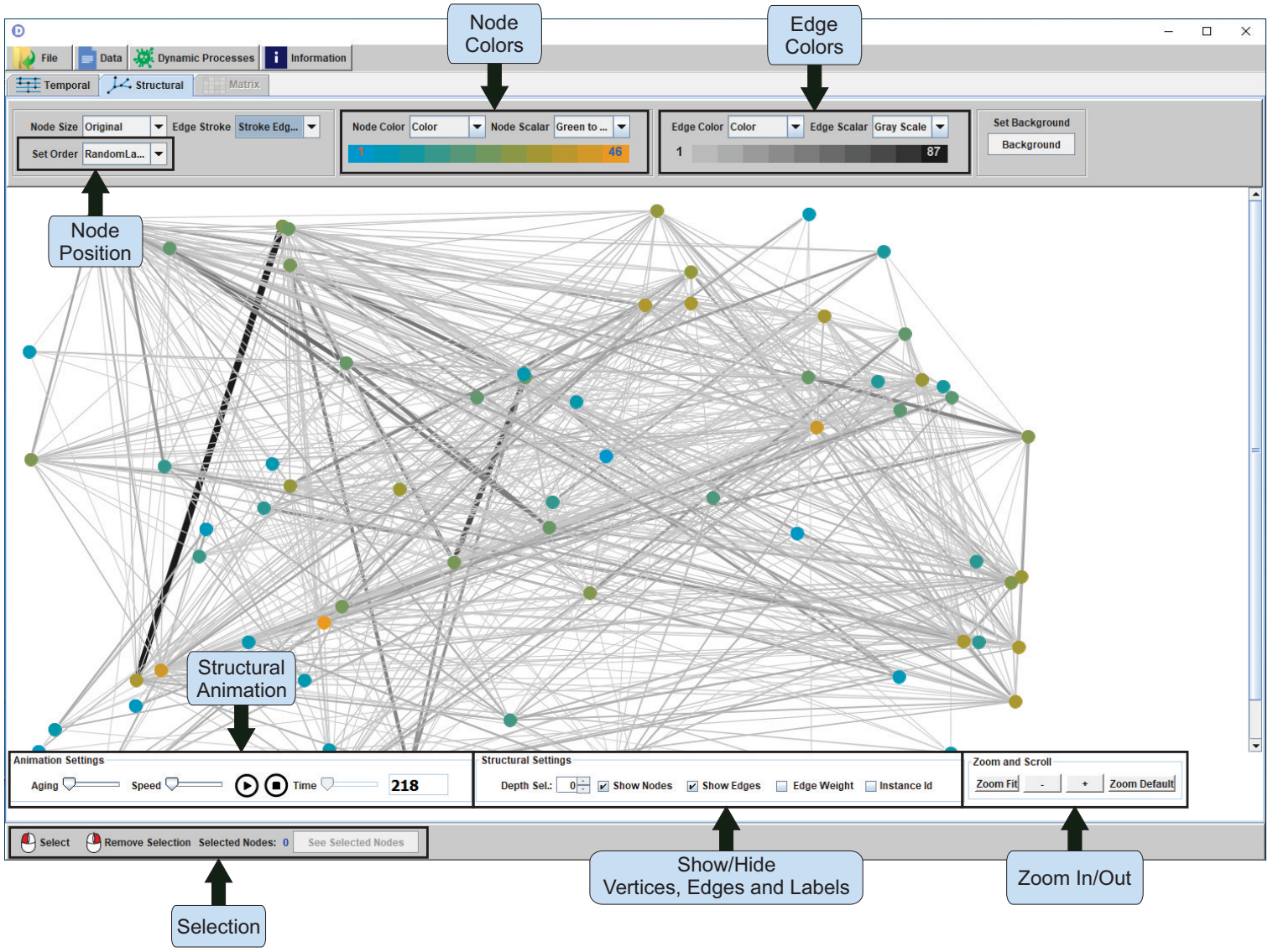

Figure 64 – Structural layout of DyNetVis demonstrating the system features.

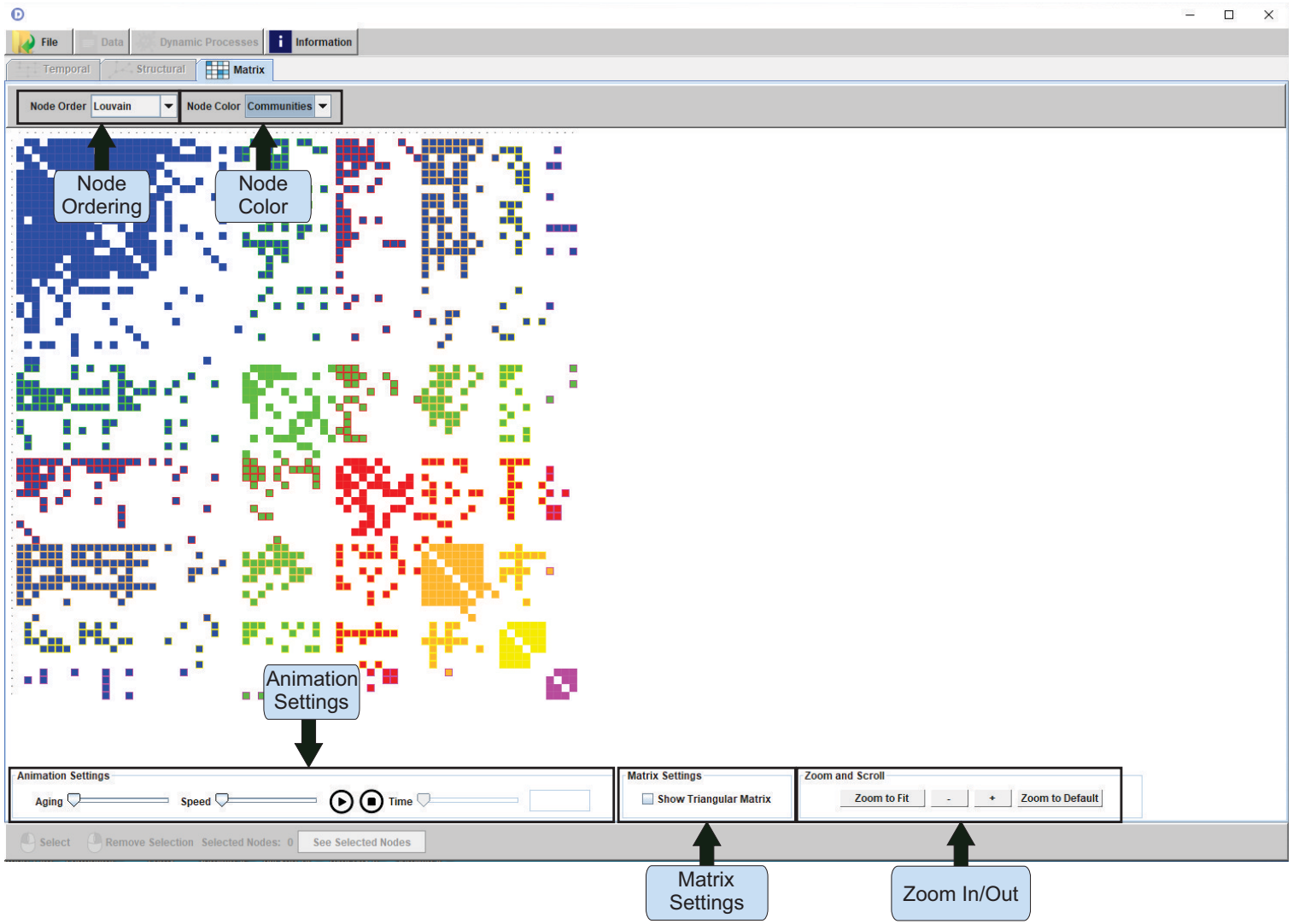

Figure 65 – Matrix layout of DyNetVis demonstrating the system features.

7.5 Final considerations

Over the past years several visualization techniques focused on temporal networks were created. However, finding an appropriate tool that includes techniques for each task is challenging and represents an open research question. This thesis presented *Dynamic Network Visualization (DyNetVis)*, an interactive software that provides several state-of-the-art visualization techniques for data analysis. DyNetVis provides tools for understanding structural and temporal patterns, including the study of the relation among network elements, the identification of abnormal behavior over time, and understanding of dynamic processes involving information spread over time.

Chapter 8

Conclusion

The visualization of temporal networks enhances the analysis of such networks, improving the perception of patterns, trends, and anomalies that would be difficult to see without the visual techniques. Due to a large amount of information on these networks, more attention has been given to issues related to the visual scalability associated with the produced layouts, but it still represents an unsolved problem and lacks effective methods.

Community detection methods are well established in the literature and are often used in large temporal networks. However, the choice of which one to use may not be trivial. We propose a strategy for comparison between two community detection methods that use visual analysis to help in this choice. Four different case studies involving two methods, Louvain and Infomap, were conducted in order to identify similarities and differences among them, considering both statistical measures and visual analysis. Our evaluation demonstrated the importance of visual analysis, since the statistics by itself represents a "black-box" and so it can be difficult for the user to decide based only on the numeric results of a network. The inclusion of the user in this process through the visualization is important for choosing the best detection algorithm for the network under analysis, enhancing the network comprehension and the perception of patterns.

We also proposed a visual scalable node reordering technique called *Community-based Node Ordering - CNO* for visual analysis of temporal networks. Our strategy improves the visual analysis of large temporal networks via a multilevel technique. We demonstrated the high quality of the CNO layout against other node reordering methods. In addition, CNO provides an edge sampling mechanism, based on the detected communities, which improves the layout analysis by allowing the perception of patterns that would otherwise be difficult to see, especially in large networks. The CNO running time is defined by the time complexity of the adopted community detection algorithm and by the node reordering techniques. Despite the chosen methods, the CNO process is flexible and allows the combination of any community detection and node reordering techniques. The experiments performed considering the quantitative and visual analysis allowed the evaluation of CNO in scenarios considering both small and large networks. The results

demonstrated coherence between the visual and quantitative analysis and allowed the identification of several patterns that would be difficult to see without the use of CNO. We have also proposed a taxonomy to categorize communities according to their activity patterns. Such taxonomy helps the user to understand the network dynamics. Moreover, it represents an efficient filter that helps the user to focus the analysis, avoiding distractions and accelerating decision making.

Another explored strategy was the use of temporal activity map to visualize dynamic processes. This strategy showed random walker trajectories (e.g., being trapped between two nodes over time), and allowed mapping of the path for infection transmissions, noticing the trajectories and timings of the infection events. Furthermore, we conducted a user evaluation to understand the relation between users' tasks and layouts, aiming to find the best layouts for respective tasks. We discovered some clues about this relation, mainly about the worst layouts for each task. Finally, to summarize all techniques and discoveries of this thesis, we created an interactive software to visualize temporal networks called *Dynamic Network Visualization (DyNetVis)*, which includes several state-of-the-art interactive visualization techniques.

8.1 Contributions of the thesis

The research conducted in this thesis created a set of novelty contributions to the fields of network science, information visualization, network communities, temporal networks, and dynamic processes. The contributions are briefly described as follows.

1. A methodology to choose the more adequate community detection algorithm supported by a visual analysis, along with a case study used four real networks, identifying similarities and differences between the algorithms both using visual and statistical measures.

2. A scalable node reordering technique for temporal networks and a taxonomy to categorize communities according to their activity patterns, tested in a case study using two real networks, improving the visual analysis of temporal networks breaking down communities and removing less relevant edges.

3. A visualization of dynamic processes for temporal networks, tested in a real network using four infection models, allowing to map the path for infection transmissions, noticing the trajectories and timings of the infection events.

4. A user evaluation to validate the relation between the users' tasks and layouts, using two real networks and four distinct layouts, finding some layouts preferences for specific tasks.

5. An interactive system to visualize temporal networks, providing state-of-the-art tools to manipulate temporal network, improving the study of the relation among network elements, the identification of abnormal behavior over time, and understanding dynamic processes involving information spread over time.

8.2 Future work

As future work for the visual analysis of community detection algorithms, we intend to improve this study with experiments based on user tasks, in order to identify whether the best method obtained by statistical measures presents the best visual experience to the user. In addition, it is also intended to extend the analysis to include other well-established community detection algorithms and also involve networks with more nodes and edges, or networks from other domains with more distinct dynamic patterns. Finally, we intend to analyze some aspects of community dynamism, such as birth, death, merge, and split of these communities (ROSSETTI; CAZABET, 2018).

Some ideas developed in this thesis may also help to visualize multiplex networks that can be seen as temporal networks containing a few snapshots. Future research efforts, however, aim to improve methods to filter edges to highlight particular temporal structures, to improve the visual analysis of larger network data sets (scalability issue) and to handle streaming networks, in which the distribution of incoming edges is unknown. The analysis of such cases may require innovative multidimensional layouts involving grouping of nodes and the complete removal of edges.

For future researches, it is possible to expand the concepts presented in this thesis for different types of networks. The applicability of the techniques in this thesis is wide open and can create new research questions in every discipline in which it is possible to represent data using temporal networks. Furthermore, efficient visualization techniques for the temporal layout, such as node ordering and edge sampling, is also an open field and requires further studies and new methods, especially related to visual scalability (LINHARES et al., 2019b).

Finally, DyNetVis software allows the integration of visualization techniques and other approaches to better visualize and explore temporal networks. Furthermore, it can be extended to implement new layouts, such as alluvial (VEHLOW et al., 2015) and circular (ELZEN et al., 2014), and also to develop new techniques for each layout. All these extensions can also generate new studies and, consequently, new research questions.

8.3 Bibliographical publications

This section contains the list of our publications and submissions with content related to this thesis. In paper [1] it is proposed the DyNetVis system, containing the reordering

algorithm *Recurrent Neighbors*, along with the Temporal Activity Map proposal, using two networks (Museum and Enron) to validate the results, described in Chapter 7. In [2] it is described as a study case using the *Recurrent Neighbors* ordering in a hospital context, detailed in Chapter 4 in the experimental results related to RN. The paper [3] proposes the CNO algorithm and the community taxonomy, with case studies for the Hospital and Twitter network, detailed in Chapter 4. The published book chapter [4] contains the visualization of dynamic processes, detailed in Chapter 5. In [5] we present a visualization method that allows the analysis of two community detection algorithms in four different networks, detailed in Chapter 3.

In [6] we present the DyNetVis software created to summarize all the contributions of this thesis, detailed in Chapter 7. Other collaborations related to social networks analysis resulted in the publications of [7,8], with studies of user preferences evolution over time, considering the social influence, using similarity networks and visualizations. Other submitted papers are related to Visualization of Streaming Networks, i.e., networks in which the edges and nodes are continuously added with unknown distribution, contain a proposal to adapt the network temporal resolution scale in an automatic fashion [9] and a streaming edge sampling technique to discard less relevant edges while maintaining the main characteristics of the network [10]. Another submitted article is related to case studies involving temporal networks applied in the educational field [11].

Thesis Publications:

1. **Linhares, C. D. G.**, Rocha, L. E. C., Paiva, J. G. S., and Travençolo, B. A. N. *DyNetVis: A system for visualization of dynamic networks.* In: Proceedings of the Symposium on Applied Computing, 2017.

2. **Linhares, C. D. G.**, Ponciano, J.R., Pereira, F. S. F., Rocha, L. E. C., Paiva, J. G. S., and Travençolo, B. A. N. *Análise temporal de uma rede de contato hospitalar utilizando técnicas de visualização de informação.* In: [S.l.]: XVII Workshop de Informática Médica, 2017.

3. **Linhares, C. D. G.**, Ponciano, J.R., Pereira, F. S. F., Rocha, L. E. C., Paiva, J. G. S., and Travençolo, B. A. N. *A Scalable Node Ordering Strategy Based on Community Structure for Enhanced Temporal Network Visualization.* In: Computers & Graphics Journal, 2019.

4. **Linhares, C. D. G.**, Ponciano, J.R., Paiva, J. G. S., Travençolo, B. A. N. and Rocha, L. E. C. *Visualisation of Structure and Processes on Temporal Networks.* In: Temporal Network Theory, Cham: Springer International Publishing, Switzerland AG, 2019.

5. **Linhares, C. D. G.**, Ponciano, J.R., Pereira, F. S. F., Rocha, L. E. C., Paiva, J. G. S., and Travençolo, B. A. N. *Visual analysis for evaluation of community detection algorithms.* In: Multimedia Tools and Applications, 2020 (to appear).

Thesis Submissions:

6. **Linhares, C. D. G.**, Ponciano, J.R., Paiva, J. G. S., Travençolo, B. A. N. and Rocha, L. E. C. *DyNetVis – An interactive software to visualize temporal networks.* In: SoftwareX, to be submitted in 03/2020.

Collaborations publications:

7. Pereira, F. S. F., **Linhares, C. D. G.**, Ponciano, J. R., Gama, J.,; de Amo, S., and Oliveira, G. M. B. *That's my jam! Uma Análise Temporal sobre a Evolução das Preferências dos Usuários em uma Rede Social de Músicas.* In: Brazilian Workshop on Social Network Analysis and Mining (Brasnam), 2018.

8. Pereira, F. S. F., **Linhares, C. D. G.**, Ponciano, J. R., Gama, J., de Amo, S., and Oliveira, G. M. B. *On Analyzing User Musical Preferences Evolution through Temporal Similarity Networks.* In: iSys - Revista Brasileira de Sistemas de Informação, 2018.

Collaborations submissions:

9. Ponciano, J. R., **Linhares, C. D. G.**, Rocha, L. E. C., Faria, E. R., and Travençolo, B. A. N. *Adaptive Temporal Resolution for Network Visualization.* In: ACM Transactions on Knowledge Discovery from Data, submitted in 01/2020.

10. Ponciano, J. R., **Linhares, C. D. G.**, Rocha, L. E. C., Faria, E. R., and Travençolo, B. A. N. *A Streaming Edge Sampling Method for Network Visualization.* In: Data Mining and Knowledge Discovery, submitted in 01/2020.

11. Ponciano, J. R., **Linhares, C. D. G.**, Melo, S. L., Lima, L. V., and Travençolo, B. A. N. *Visual analysis of contact patterns in school environments.* In: Informatics in Education, submitted in 12/2019.

References

ABELLO, J.; HAM, F. V.; KRISHNAN, N. Ask-graphview: A large scale graph visualization system. **IEEE Transactions on Visualization and Computer Graphics**, v. 12, n. 5, p. 669–676, 2006. Disponível em: <https://doi.org/10.1109/TVCG.2006.120>.

AHN, J.-W.; PLAISANT, C.; SHNEIDERMAN, B. A task taxonomy for network evolution analysis. **IEEE Transactions on Visualization and Computer Graphics**, IEEE Educational Activities Department, Piscataway, NJ, USA, v. 20, n. 3, p. 365–376, mar. 2014. ISSN 1077-2626. Disponível em: <https://doi.org/10.1109/TVCG.2013.238>.

AHN, Y.-Y.; BAGROW, J. P.; LEHMANN, S. Link communities reveal multiscale complexity in networks. **Nature**, Nature Publishing Group, v. 466, n. 7307, p. 761–764, 2010. Disponível em: <https://doi.org/10.1038/nature09182>.

ARCHAMBAULT, D.; PURCHASE, H. C. Can animation support the visualisation of dynamic graphs? **Information Sciences**, v. 330, p. 495 – 509, 2016. ISSN 0020-0255. Disponível em: <https://doi.org/10.1016/j.ins.2015.04.017>.

BACH, B. Unfolding dynamic networks for visual exploration. **IEEE Computer Graphics and Applications**, v. 36, n. 2, p. 74–82, Mar 2016. ISSN 0272-1716. Disponível em: <https://doi.org/10.1109/MCG.2016.32>.

BACH, B.; PIETRIGA, E.; FEKETE, J.-D. GraphDiaries: Animated Transitions and Temporal Navigation for Dynamic Networks. **IEEE Transactions on Visualization and Computer Graphics**, Institute of Electrical and Electronics Engineers, v. 20, n. 5, p. 740 – 754, maio 2014. Disponível em: <https://doi.org/10.1109/TVCG.2013.254>.

BARRAT, A.; BARTHéLEMY, M.; VESPIGNANI, A. **Dynamical Processes on Complex Networks**. Cambridge University Press, 2008. Disponível em: <https://doi.org/10.1017/CBO9780511791383>.

BASSETT, D. S. et al. Robust detection of dynamic community structure in networks. **CoRR**, abs/1206.4358, 2012. Disponível em: <https://doi.org/10.1063/1.4790830>.

BASTIAN, M.; HEYMANN, S.; JACOMY, M. Gephi: An open source software for exploring and manipulating networks. In: **ICWSM**. [s.n.], 2009. Disponível em: <https://doi.org/10.13140/2.1.1341.1520>.

BATTISTA, G. D. et al. Algorithms for drawing graphs: an annotated bibliography. **Computational Geometry**, v. 4, n. 5, p. 235 – 282, 1994. Disponível em: <https://doi.org/10.1016/0925-7721(94)00014-X>.

BECK, F.; BURCH, M.; DIEHL, S. Matching application requirements with dynamic graph visualization profiles. In: **2013 17th International Conference on Information Visualisation**. [s.n.], 2013. p. 11–18. ISSN 1550-6037. Disponível em: <https://doi.org/10.1109/IV.2013.2>.

BECK, F. et al. The State of the Art in Visualizing Dynamic Graphs. In: BORGO, R.; MACIEJEWSKI, R.; VIOLA, I. (Ed.). **EuroVis - STARs**. The Eurographics Association, 2014. ISBN 978-3-03868-028-4. Disponível em: <https://doi.org/10.2312/eurovisstar.20141174>.

_____. A taxonomy and survey of dynamic graph visualization. **Computer Graphics Forum**, v. 36, n. 1, p. 133–159, 2016. ISSN 1467-8659. Disponível em: <https://doi.org/10.1111/cgf.12791>.

BEHRISCH, M. et al. Matrix reordering methods for table and network visualization. In: WILEY ONLINE LIBRARY. **Computer Graphics Forum**. 2016. v. 35, n. 3, p. 693–716. Disponível em: <https://doi.org/10.1111/cgf.12935>.

BLONDEL, V. D. et al. Fast unfolding of communities in large networks. **Journal of Statistical Mechanics: Theory and Experiment**, v. 2008, n. 10, p. P10008, October 2008. Disponível em: <https://doi.org/10.1088/1742-5468/2008/10/p10008>.

BOYANDIN, I.; BERTINI, E.; LALANNE, D. A qualitative study on the exploration of temporal changes in flow maps with animation and small-multiples. **Computer Graphics Forum**, v. 31, n. 3pt2, p. 1005–1014, 2012. Disponível em: <https://doi.org/10.1111/j.1467-8659.2012.03093.x>.

BRANDES, U. Force-directed graph drawing. In: _____. **Encyclopedia of Algorithms**. Boston, MA: Springer US, 2008. p. 1–6. ISBN 978-3-642-27848-8. Disponível em: <https://doi.org/10.1007/978-3-642-27848-8_648-1>.

BURCH, M. Visual analytics of large dynamic digraphs. **Information Visualization**, v. 16, n. 3, p. 167–178, 2017. Disponível em: <https://doi.org/10.1177/1473871616661194>.

CARRASCOSA, J. M. et al. Are trending topics useful for marketing?: Visibility of trending topics vs traditional advertisement. In: **Proceedings of the First ACM Conference on Online Social Networks**. New York, NY, USA: ACM, 2013. (COSN '13), p. 165–176. ISBN 978-1-4503-2084-9. Disponível em: <https://doi.org/10.1145/2512938.2512948>.

CATTUTO, C. et al. Dynamics of person-to-person interactions from distributed rfid sensor networks. **PloS one**, Public Library of Science, v. 5, n. 7, p. e11596, 2010. Disponível em: <https://doi.org/10.1371/journal.pone.0011596>.

CLARK-CARTER, D. Interquartile range. In: _____. **Encyclopedia of Statistics in Behavioral Science**. American Cancer Society, 2005. ISBN 9780470013199. Disponível em: <https://doi.org/10.1002/0470013192.bsa311>.

COL, A. D. et al. Wavelet-based visual analysis of dynamic networks. **IEEE transactions on visualization and computer graphics**, v. 24, n. 8, p. 2456—2469, August 2018. ISSN 1077-2626. Disponível em: <https://doi.org/10.1109/TVCG.2017.2746080>.

CORNELISSEN, B. et al. Understanding execution traces using massive sequence and circular bundle views. In: **ICPC**. IEEE Computer Society, 2007. p. 49–58. Disponível em: <https://doi.org/10.1109/ICPC.2007.39>.

COSTA, L. da F. et al. Analyzing and modeling real-world phenomena with complex networks: a survey of applications. **Advances in Physics**, Taylor & Francis, v. 60, n. 3, p. 329–412, 2011. Disponível em: <https://doi.org/10.1080/00018732.2011.572452>.

CRAMPES, M.; PLANTIÉ, M. A unified community detection, visualization and analysis method. **Advances in Complex Systems**, v. 17, 2014. Disponível em: <https://doi.org/10.1142/S0219525914500015>.

DRIF, A.; BOUKERRAM, A. Taxonomy and survey of community discovery methods in complex networks. **International Journal of Computer Science and Engineering Survey**, Academy & Industry Research Collaboration Center (AIRCC), v. 5, n. 4, p. 1, 2014. Disponível em: <https://doi.org/10.5121/ijcses.2014.5401>.

DUNNE, C.; SHNEIDERMAN, B. Motif simplification: Improving network visualization readability with fan, connector, and clique glyphs. In: **Proceedings of the SIGCHI Conference on Human Factors in Computing Systems**. New York, NY, USA: ACM, 2013. (CHI '13), p. 3247–3256. ISBN 978-1-4503-1899-0. Disponível em: <https://doi.org/10.1145/2470654.2466444>.

ELMQVIST, N. et al. Zame: Interactive large-scale graph visualization. In: **2008 IEEE Pacific Visualization Symposium**. [s.n.], 2008. p. 215–222. ISSN 2165-8765. Disponível em: <https://doi.org/10.1109/PACIFICVIS.2008.4475479>.

ELZEN, S. van den et al. Reordering massive sequence views: Enabling temporal and structural analysis of dynamic networks. In: IEEE. **Visualization Symposium (PacificVis), 2013 IEEE Pacific**. 2013. p. 33–40. Disponível em: <https://doi.org/10.1109/PacificVis.2013.6596125>.

_____. Dynamic network visualization with extended massive sequence views. **IEEE Transactions on Visualization and Computer Graphics**, v. 20, p. 1087–1099, 2014. Disponível em: <https://doi.org/10.1109/TVCG.2013.263>.

_____. Reducing snapshots to points: A visual analytics approach to dynamic network exploration. **IEEE Transactions on Visualization and Computer Graphics**, v. 22, n. 1, p. 1–10, Jan 2016. ISSN 1077-2626. Disponível em: <https://doi.org/10.1109/TVCG.2015.2468078>.

FORTUNATO, S. Community detection in graphs. **Physics Reports**, v. 486, n. 3, p. 75 – 174, 2010. ISSN 0370-1573. Disponível em: <https://doi.org/10.1016/j.physrep.2009.11.002>.

FORTUNATO, S.; BARTHLEMY, M. Resolution limit in community detection. **Proceedings of the National Academy of Sciences**, v. 104, n. 1, p. 36–41, 2007. Disponível em: <https://doi.org/10.1073/pnas.0605965104>.

FORTUNATO, S.; HRIC, D. Community detection in networks: A user guide. **Physics Reports**, v. 659, n. Supplement C, p. 1 – 44, 2016. ISSN 0370-1573. Disponível em: <https://doi.org/10.1016/j.physrep.2016.09.002>.

GEMMETTO, V.; BARRAT, A.; CATTUTO, C. Mitigation of infectious disease at school: targeted class closure vs school closure. **BMC infectious diseases**, BioMed Central Ltd, v. 14, n. 1, p. 695, dez. 2014. ISSN 1471-2334. Disponível em: <https://doi.org/10.1186/s12879-014-0695-9>.

GÉNOIS, M. et al. Data on face-to-face contacts in an office building suggest a low-cost vaccination strategy based on community linkers. **Network Science**, v. 3, p. 326–347, 2015. Disponível em: <https://doi.org/10.1017/nws.2015.10>.

GHONIEM, M.; FEKETE, J.-D.; CASTAGLIOLA, P. A comparison of the readability of graphs using node-link and matrix-based representations. In: **IEEE Symposium on Information Visualization**. [s.n.], 2004. p. 17–24. ISSN 1522-404X. Disponível em: <https://doi.org/10.1109/INFVIS.2004.1>.

_____. On the readability of graphs using node-link and matrix-based representations: A controlled experiment and statistical analysis. **Information Visualization**, v. 4, p. 114 – 135, 2005. Disponível em: <https://doi.org/10.1057/palgrave.ivs.9500092>.

GHONIEM, M. et al. Vafle: visual analytics of firewall log events. **Proc.SPIE**, v. 9017, February 2014. Disponível em: <https://doi.org/10.1117/12.2037790>.

GIALAMPOUKIDIS, I. et al. Community detection in complex networks based on dbscan* and a martingale process. In: IEEE. **Semantic and Social Media Adaptation and Personalization (SMAP), 2016 11th International Workshop on**. 2016. p. 1–6. Disponível em: <https://doi.org/10.1109/SMAP.2016.7753375>.

GóMEZ, S. et al. Discrete-time markov chain approach to contact-based disease spreading in complex networks. **EPL (Europhysics Letters)**, IOP Publishing, v. 89, n. 3, p. 38009, Feb 2010. ISSN 1286-4854. Disponível em: <https://doi.org/10.1209/0295-5075/89/38009>.

HARPE, S. E. How to analyze likert and other rating scale data. **Currents in Pharmacy Teaching and Learning**, v. 7, n. 6, p. 836 – 850, 2015. ISSN 1877-1297. Disponível em: <https://doi.org/10.1016/j.cptl.2015.08.001>.

HOLTEN, D.; WIJK, J. J. V. Force-directed edge bundling for graph visualization. **Computer Graphics Forum**, v. 28, n. 3, p. 983–990, 2009. Disponível em: <https://doi.org/10.1111/j.1467-8659.2009.01450.x>.

HUANG, W.; EADESBAND, P.; HONG, S.-H. Measuring effectiveness of graph visualizations: A cognitive load perspective. **Information Visualization**, v. 8, n. 3, p. 139–152, 2009. Disponível em: <https://doi.org/10.1057/ivs.2009.10>.

IMRAN, M. et al. Aidr: Artificial intelligence for disaster response. In: **Proceedings of the 23rd International Conference on World Wide Web**. New York, NY, USA: ACM, 2014. (WWW '14 Companion), p. 159–162. ISBN 978-1-4503-2745-9. Disponível em: <https://doi.org/10.1145/2567948.2577034>.

ISELLA, L. et al. What's in a crowd? analysis of face-to-face behavioral networks. **J Theor Biol**, Elsevier, v. 271, n. 1, p. 166–180, 2011. Disponível em: <https://doi.org/10.1016/j.jtbi.2010.11.033>.

KAPANIPATHI, P. et al. Hierarchical interest graph from tweets. In: **Proceedings of the 23rd International Conference on World Wide Web**. New York, NY, USA: ACM, 2014. (WWW '14 Companion), p. 311–312. ISBN 978-1-4503-2745-9. Disponível em: <https://doi.org/10.1145/2567948.2577353>.

KEILA, P. S.; SKILLICORN, D. B. Structure in the enron email dataset. **Computational & Mathematical Organization Theory**, 2005. Disponível em: <https://doi.org/10.1007/s10588-005-5379-y>.

KEIM, D. Visual exploration of large data sets. **Communications of the ACM**, v. 44, n. 8, p. 38–44, 2001. Disponível em: <https://doi.org/10.1145/381641.381656>.

KEIM, D. A. Information visualization and visual data mining. **IEEE Transactions on Visualization and Computer Graphics**, IEEE Educational Activities Department, USA, v. 8, n. 1, p. 1–8, jan. 2002. ISSN 1077-2626. Disponível em: <https://doi.org/10.1109/2945.981847>.

KELLER, R.; ECKERT, C. M.; CLARKSON, P. J. Matrices or node-link diagrams: Which visual representation is better for visualising connectivity models? **Information Visualization**, v. 5, n. 1, p. 62–76, 2006. Disponível em: <https://doi.org/10.1057/palgrave.ivs.9500116>.

KERRACHER, N. et al. Visual Techniques to Support Exploratory Analysis of Temporal Graph Data. In: BERTINI, E.; KENNEDY, J.; PUPPO, E. (Ed.). **Eurographics Conference on Visualization (EuroVis) - Short Papers**. The Eurographics Association, 2015. Disponível em: <https://doi.org/10.2312/eurovisshort.20151133>.

KYROLA, A.; BLELLOCH, G.; GUESTRIN, C. Graphchi: Large-scale graph computation on just a pc. In: **Proceedings of the 10th USENIX Conference on Operating Systems Design and Implementation**. Berkeley, CA, USA: USENIX Association, 2012. (OSDI'12), p. 31–46. ISBN 978-1-931971-96-6. Disponível em: <https://doi.org/10.21236/ada603410>.

LAMBERT, A.; BOURQUI, R.; AUBER, D. 3d edge bundling for geographical data visualization. In: **2010 14th International Conference Information Visualisation**. [s.n.], 2010. p. 329–335. ISSN 2375-0138. Disponível em: <https://doi.org/10.1109/IV.2010.53>.

LANCICHINETTI, A.; FORTUNATO, S. Community detection algorithms: A comparative analysis. **Physical Review E**, American Physical Society, v. 80, p. 056117, Nov 2009. Disponível em: <https://doi.org/doi/10.1103/PhysRevE.80.056117>.

LHUILLIER, A.; HURTER, C.; TELEA, A. Ffteb: Edge bundling of huge graphs by the fast fourier transform. In: **2017 IEEE Pacific Visualization Symposium (PacificVis)**. [s.n.], 2017. p. 190–199. ISSN 2165-8773. Disponível em: <https://doi.org/10.1109/PACIFICVIS.2017.8031594>.

LI, M. et al. DyNetViewer: a Cytoscape app for dynamic network construction, analysis and visualization. **Bioinformatics**, v. 34, n. 9, p. 1597–1599, 12 2017. ISSN 1367-4803. Disponível em: <https://doi.org/10.1093/bioinformatics/btx821>.

LIM, K. H.; DATTA, A. Following the follower: Detecting communities with common interests on twitter. In: **Proceedings of the 23rd ACM Conference on Hypertext and Social Media**. New York, NY, USA: ACM, 2012. (HT '12), p. 317–318. ISBN 978-1-4503-1335-3. Disponível em: <https://doi.org/10.1145/2309996.2310052>.

LIN, Y.-R. et al. Facetnet: a framework for analyzing communities and their evolutions in dynamic networks. In: ACM. **Proceedings of the 17th international conference on World Wide Web**. 2008. p. 685–694. Disponível em: <https://doi.org/10.1145/1367497.1367590>.

LINHARES, C. D. G. et al. Visualisation of structure and processes on temporal networks. In: _____. **Temporal Network Theory**. Cham: Springer International Publishing, 2019. p. 83–105. ISBN 978-3-030-23495-9. Disponível em: <https://doi.org/10.1007/978-3-030-23495-9_5>.

_____. A scalable node ordering strategy based on community structure for enhanced temporal network visualization. **Computers & Graphics**, v. 84, p. 185 – 198, 2019. ISSN 0097-8493. Disponível em: <https://doi.org/10.1016/j.cag.2019.08.006>.

_____. Visual analysis for evaluation of community detection algorithms. **Multimedia Tools and Applications**, 2020. Disponível em: <https://doi.org/10.1007/s11042-020-08700-4>.

_____. Análise temporal de uma rede de contato hospitalar utilizando técnicas de visualização de informação. In: . XVII Workshop de Informática Médica, 2017. Disponível em: <https://doi.org/10.5753/sbcas.2017.3696>.

_____. DyNetVis: a system for visualization of dynamic networks. In: **Proceedings of the Symposium on Applied Computing**. Marrakech, Morocco: ACM, 2017. (SAC '17), p. 187–194. ISBN 978-1-4503-4486-9. Disponível em: <https://doi.org/10.1145/3019612.3019686>.

MALEWICZ, G. et al. Pregel: A system for large-scale graph processing. In: **Proceedings of the 2010 ACM SIGMOD International Conference on Management of Data**. New York, NY, USA: ACM, 2010. (SIGMOD '10), p. 135–146. ISBN 978-1-4503-0032-2. Disponível em: <https://doi.org/10.1145/1807167.1807184>.

MASTRANDREA, R.; FOURNET, J.; BARRAT, A. Contact patterns in a high school: A comparison between data collected using wearable sensors, contact diaries and friendship surveys. **PLOS ONE**, Public Library of Science, v. 10, n. 9, p. 1–26, 09 2015. Disponível em: <https://doi.org/10.1371/journal.pone.0136497>.

MASUDA, N.; LAMBIOTTE, R. **A Guide to Temporal Networks**. World Scientific, 2016. Disponível em: <https://doi.org/10.1142/q0033>.

MI, P. et al. Interactive graph layout of a million nodes. **Informatics**, v. 3, p. 23, 12/2016 2016. Disponível em: <https://doi.org/10.3390/informatics3040023>.

MOTHE, J.; MKHITARYAN, K.; HAROUTUNIAN, M. Community detection: Comparison of state of the art algorithms. In: **2017 Computer Science and Information Technologies (CSIT)**. [s.n.], 2017. p. 125–129. Disponível em: <https://doi.org/10.1109/CSITechnol.2017.8312155>.

NEWMAN, M. E. J. Equivalence between modularity optimization and maximum likelihood methods for community detection. **Phys. Rev. E**, American Physical Society, v. 94, p. 052315, Nov 2016. Disponível em: <https://doi.org/10.1103/PhysRevE.94.052315>.

NIELSEN, J.; MOLICH, R. Heuristic evaluation of user interfaces. In: **Proceedings of the SIGCHI Conference on Human Factors in Computing Systems**. New York, NY, USA: ACM, 1990. (CHI '90), p. 249–256. ISBN 0-201-50932-6. Disponível em: <https://doi.org/10.1145/97243.97281>.

ONNELA, J.-P. et al. Taxonomies of networks from community structure. **Physical Review E**, APS, v. 86, n. 3, p. 036104, 2012. Disponível em: <https://doi.org/10.1103/PhysRevE.86.036104>.

ORMAN, G. K.; LABATUT, V.; CHERIFI, H. Comparative evaluation of community detection algorithms: a topological approach. **Journal of Statistical Mechanics: Theory and Experiment**, IOP Publishing, v. 2012, n. 08, p. P08001, aug 2012. Disponível em: <https://doi.org/10.1088/1742-5468/2012/08/p08001>.

Orman, G. K. et al. A method for characterizing communities in dynamic attributed complex networks. In: **2014 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM 2014)**. [s.n.], 2014. p. 481–484. Disponível em: <https://doi.org/10.1109/ASONAM.2014.6921629>.

PEREIRA, F. S.; AMO, S. d.; GAMA, J. Detecting events in evolving social networks through node centrality analysis. In: **Workshop on Large-scale Learning from Data Streams in Evolving Environments of ECML PKDD**. [S.l.: s.n.], 2016. p. 83–93.

PERER, A.; SHNEIDERMAN, B. Integrating statistics and visualization: Case studies of gaining clarity during exploratory data analysis. In: **Proceedings of the SIGCHI Conference on Human Factors in Computing Systems**. New York, NY, USA: ACM, 2008. (CHI '08), p. 265–274. ISBN 978-1-60558-011-1. Disponível em: <https://doi.org/10.1145/1357054.1357101>.

RAJPOOT, K. et al. Functional connectivity alterations in epilepsy from resting-state functional mri. **PloS one**, v. 10, p. e0134944, 08 2015. Disponível em: <https://doi.org/10.1371/journal.pone.0134944>.

RIBEIRO, B.; PERRA, N.; BARONCHELLI, A. Quantifying the effect of temporal resolution on time-varying networks. **Scientific reports**, Nature Publishing Group, v. 3, p. 3006, 2013. Disponível em: <https://doi.org/10.1038/srep03006>.

ROCHA, L. E. C.; BLONDEL, V. D. Bursts of vertex activation and epidemics in evolving networks. **PLOS Computational Biology**, v. 9, p. e1002974, 2013. Disponível em: <https://doi.org/10.1371/journal.pcbi.1002974>.

ROCHA, L. E. C.; LILJEROS, F.; HOLME, P. Simulated epidemics in an empirical spatiotemporal network of 50,185 sexual contacts. **PLoS Computational Biology**, Public Library of Science, v. 7, n. 3, p. e1001109, 03 2011. Disponível em: <https://doi.org/10.1371/journal.pcbi.1001109>.

ROCHA, L. E. C.; MASUDA, N. Random walk centrality for temporal networks. **New Journal of Physics**, v. 16, p. 063023, 2014. Disponível em: <https://doi.org/10.1088/1367-2630/16/6/063023>.

ROCHA, L. E. C.; MASUDA, N.; HOLME, P. Sampling of temporal networks: Methods and biases. **Phys. Rev. E**, American Physical Society, v. 96, p. 052302, Nov 2017. Disponível em: <https://doi.org/10.1103/PhysRevE.96.052302>.

ROSSETTI, G.; CAZABET, R. Community discovery in dynamic networks: A survey. **ACM Computing Surveys**, v. 51, n. 2, p. 35, 2018. Disponível em: <https://doi.org/10.1145/3172867>.

ROSVALL, M.; BERGSTROM, C. T. Maps of random walks on complex networks reveal community structure. **Proceedings of the National Academy of Sciences**, v. 105, n. 4, p. 1118–1123, 2008. Disponível em: <https://doi.org/10.1073/pnas.0706851105>.

_____. Mapping change in large networks. **PLoS ONE**, Public Library of Science, v. 5, n. 1, p. e8694, 01 2010. Disponível em: <https://doi.org/10.1371/journal.pone.0008694>.

ROSVALL, M. et al. Different approaches to community detection. In: _____. **Advances in Network Clustering and Blockmodeling**. John Wiley & Sons, Ltd, 2019. cap. 4, p. 105–119. ISBN 9781119483298. Disponível em: <https://doi.org/10.1002/9781119483298.ch4>.

SALLABERRY, A.; MUELDER, C.; MA, K.-L. Clustering, visualizing, and navigating for large dynamic graphs. In: DIDIMO, W.; PATRIGNANI, M. (Ed.). **Graph Drawing**. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013. p. 487–498. ISBN 978-3-642-36763-2. Disponível em: <https://doi.org/10.1007/978-3-642-36763-2_43>.

SCHOLTES, I. et al. Causality-driven slow-down and speed-up of diffusion in non-markovian temporal networks. **Nature Communications**, v. 5, p. 5024, 2014. Disponível em: <https://doi.org/10.1038/ncomms6024>.

SHANNON, P. et al. Cytoscape: a software environment for integrated models of biomolecular interaction networks. **Genome Research**, v. 13, n. 11, p. 2498–2504, nov. 2003. Disponível em: <https://doi.org/10.1101/gr.1239303>.

SHARMA, A. Modeling the effect of people's preferences and social forces on adopting and sharing items. In: **Proceedings of the 8th ACM Conference on Recommender Systems**. New York, NY, USA: ACM, 2014. (RecSys '14), p. 421–424. ISBN 978-1-4503-2668-1. Disponível em: <https://doi.org/10.1145/2645710.2653364>.

SHNEIDERMAN, B. The eyes have it: a task by data type taxonomy for information visualizations. In: **Proceedings 1996 IEEE Symposium on Visual Languages**. [s.n.], 1996. p. 336–343. ISSN 1049-2615. Disponível em: <https://doi.org/10.1109/VL.1996.545307>.

SIX, J. M.; TOLLIS, I. G. A framework and algorithms for circular drawings of graphs. **Journal of Discrete Algorithms**, v. 4, n. 1, p. 25 – 50, March 2006. ISSN 1570-8667. Disponível em: <https://doi.org/10.1016/j.jda.2005.01.009>.

STARNINI, M. et al. Random walks on temporal networks. **Phys. Rev. E**, v. 85, p. 056115, 2012. Disponível em: <https://doi.org/10.1103/PhysRevE.85.056115>.

STEHLÉ, J. et al. High-resolution measurements of face-to-face contact patterns in a primary school. **PLOS ONE**, Public Library of Science, v. 6, n. 8, p. e23176, 08 2011. Disponível em: <https://doi.org/10.1371/journal.pone.0023176>.

TANAHASHI, Y.; MA, K. L. Design considerations for optimizing storyline visualizations. **IEEE Transactions on Visualization and Computer Graphics**, v. 18, n. 12, p. 2679–2688, Dec 2012. ISSN 1077-2626. Disponível em: <https://doi.org/10.1109/TVCG.2012.212>.

TERGAN, S.-O.; KELLER, T. **Knowledge and Information Visualization**. Berlin, Heidelberg: Springer-Verlag, 2005. ISBN 3540269215. Disponível em: <https://doi.org/10.1007/b138081>.

TRAUD, A. L. et al. Visualization of communities in networks. **Chaos: An Interdisciplinary Journal of Nonlinear Science**, v. 19, n. 4, p. 041104, 2009. Disponível em: <https://doi.org/10.1063/1.3194108>.

VANHEMS, P. et al. Estimating potential infection transmission routes in hospital wards using wearable proximity sensors. **PLoS One**, v. 8, p. e73970, 2013. Disponível em: <https://doi.org/10.1371/journal.pone.0073970>.

VEHLOW, C. et al. Visualizing the evolution of communities in dynamic graphs. **Computer Graphics Forum**, v. 34, n. 1, p. 277–288, 2015. ISSN 1467-8659. Disponível em: <https://doi.org/10.1111/cgf.12512>.

WANG, W.; STREET, W. N. A novel algorithm for community detection and influence ranking in social networks. In: IEEE. **Advances in Social Networks Analysis and Mining (ASONAM), 2014 IEEE/ACM International Conference on**. 2014. p. 555–560. Disponível em: <https://doi.org/10.1109/ASONAM.2014.6921641>.

WANG, W. et al. Visualization of large hierarchical data by circle packing. In: **Proceedings of the SIGCHI Conference on Human Factors in Computing Systems**. New York, NY, USA: ACM, 2006. (CHI '06), p. 517–520. ISBN 1-59593-372-7. Disponível em: <https://doi.org/10.1145/1124772.1124851>.

WARE, C. Information visualization: Perception for design. In: _____. Boston: Morgan Kaufmann, 2013. (Interactive Technologies), p. 514. ISBN 978-0-12-381464-7. Disponível em: <https://doi.org/10.1016/B978-0-12-381464-7.00018-1>.

WU, Y. et al. A survey on visual analytics of social media data. **IEEE Transactions on Multimedia**, v. 18, n. 11, p. 2135–2148, Nov 2016. ISSN 1520-9210. Disponível em: <https://doi.org/10.1109/TMM.2016.2614220>.

YANG, Z.; ALGESHEIMER, R.; TESSONE, C. J. A comparative analysis of community detection algorithms on artificial networks. **Scientific Reports**, v. 6, 2016. Disponível em: <https://doi.org/10.1038/srep30750>.

YIN, C. et al. A method for community detection of complex networks based on hierarchical clustering. **IJDSN**, v. 2015, p. 849140:1–849140:9, 2015. Disponível em: <https://doi.org/10.1155/2015/849140>.

ZHANG, Q.-G. et al. Drawing undirected graphs with genetic algorithms. In: WANG, L.; CHEN, K.; ONG, Y. (Ed.). **Advances in Natural Computation**. Springer Berlin Heidelberg, 2005, (Lecture Notes in Computer Science, v. 3612). p. 28–36. ISBN 978-3-540-28320-1. Disponível em: <https://doi.org/10.1007/11539902_4>.

ZHAO, Y. et al. Eod edge sampling for visualizing dynamic network via massive sequence view. **IEEE Access**, v. 6, p. 53006–53018, 2018. ISSN 2169-3536. Disponível em: <https://doi.org/10.1109/ACCESS.2018.2870684>.