Journal Title
XX(X):1–3

DOI: 10.1177/ToBeAssigned
www.sagepub.com/
SAGE

Communicability-derived layout and community detection in complex networks

Renato Fabbri[1] and Ernesto Estrada[2]

Abstract

This paper describes the use of the communicability measure for the achievement of layouts and community detection in complex networks.

Keywords

Complex networks, communicability, community detection, layout algorithms, interactive visualization

Introduction

Related work

The communicability framework

Preliminaries

Layouts

Community detection

Results

An interactive interface

We implemented the framework described in the previous section using scientific, database and web resources, making it available within a web page for use through simple mouse-driven actions and requiring no software installation beyond a web browser. The software has been named ComNetVis (from Communicability-based Network Visualization) and is exemplified in Figures 1, 2 and **??**. In the following subsections we describe its functionalities and the underlying technologies.

Using ComNetVis

In using ComNetVis, the analyst first uses a drop-down menu to select a network of interest, s/he may also upload a new network using the same menu. Then, s/he sets the paramenters for the calculation of the communicabiity angles, as illustrated in Figure 2, although default settings are reasonable, specially for a newcomer. The next step is to set the community detection algorithm. To do so, the user selects a dimmensionality reduction method, and a target dimmensionality in which clustering is going to be performed. Accordingly, s/he selects a clustering algorithm and the minimum and maximum number of communities of interest, as illustrated in Figure 3. The last step is to set the node-link layout algorithm: the user selects a dimmensionality reduction method, and the number of dimensions of the layout, as illustrated in Figure 4.

Theoretically, any dimensionality reduction method and any clustering algorithm can be used. In this interface, to encompass a reasonable collection of techniques, and enable the proof-of-concept, we implemented five dimmensionality reduction methods (PCA [1], t-SNE [2], Locally Linear Embedding [3], ISOMAP [4], and MDS [5]), and three clustering algorithms (k-means, Ward's hierarchical [6], and spectral [7]).

Once the "render network" button is hit, the network node-link representation is mapped to the screen according to the selected settings. Subsequent usage relies on manipulation of the visual mapping controls presented on the canvas by means of user actions and related mouse operations. These steps are illustrated in Figure 5.

Underlying software technologies

Components of ComNetVis are mostly written in a combination of JavaScript and Python: it uses Vue.js (set up by Nuxt.js) in the front-end client, the back-end is a Flask Python server, used to perform specialized or heavy calculations. A secondary server, a FeatherJS, is used to facilitate contact with the database and real-time multi-user interaction. The data is stored in a MongoDB database and ordinarily, while the multi-user interaction is deactivated for now to avoid unnecessary complexity. The communicability-related calculations for community detection and network layout are performed by the Flask server. The fast WebGL 3D rendering on the canvas is performed using Babylon.js.

Use cases

Conclusions and further work

Acknowledgements

The authors acknowledge the financial support of the São Paulo State Research Foundation (FAPESP grant 2017/05838-3). The

[1]Institute of Mathematics and Computer Sciences, University of So Paulo, So Carlos, Brazil
[2] Department of Mathematics and Statistics, University of Strathclyde, Glasgow G11HX, United Kingdom

Corresponding author:
Renato Fabbri, Institute of Mathematics and Computer Sciences, University of So Paulo, So Carlos 13566-590, Brazil.
Email: renato.fabbri@gmail.com

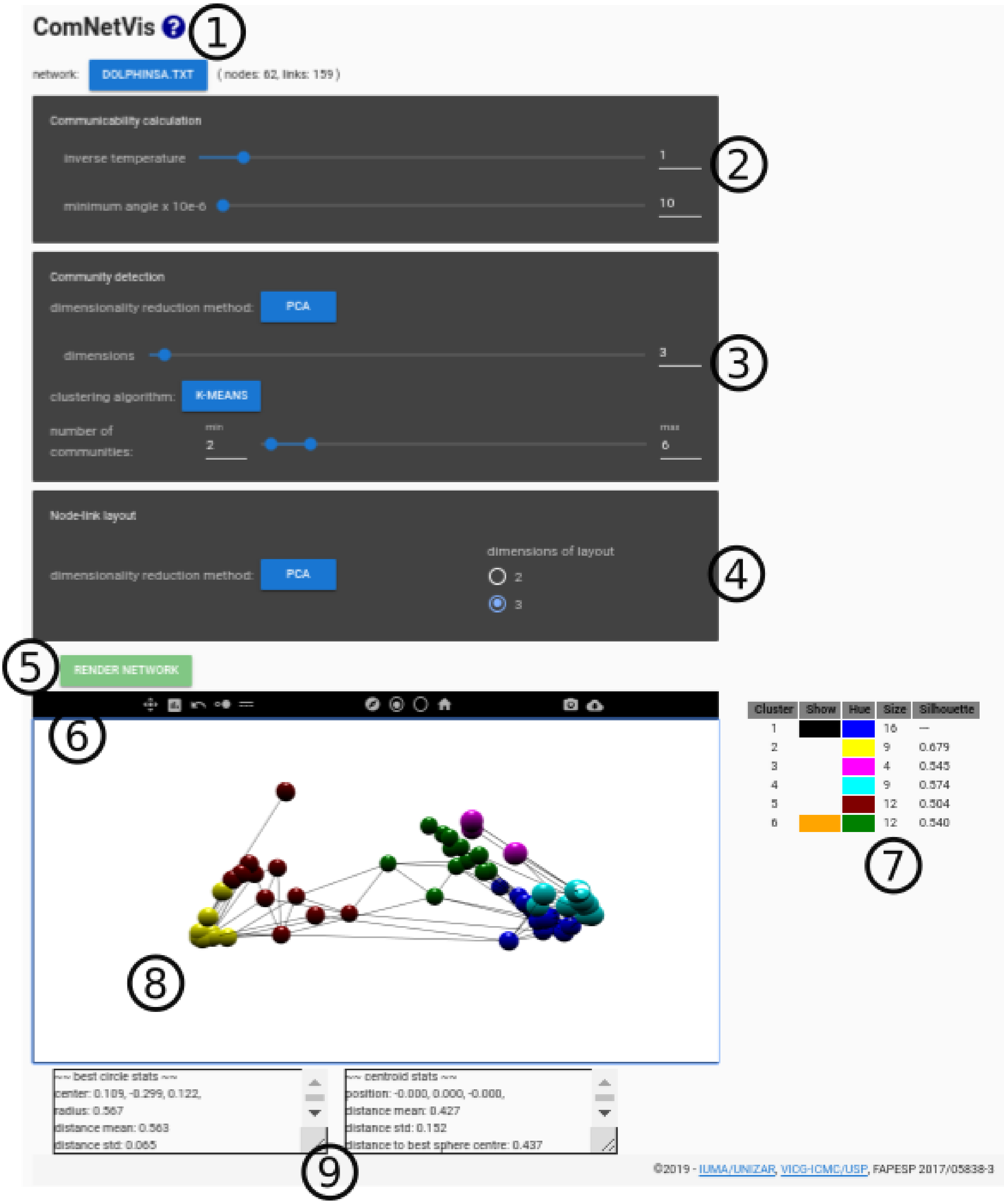

Figure 1. The ComNetVis interface: (1) drop-down menu for selecting or uploading a network for analysis; (2) widgets for setting the communicability calculations; (3) widgets for setting the community detection; (4) widgets for setting the node-link layout; (5) the render button to display the network; (6) toolbar with controls for navigation of the network and fine-tuning the visualization; (7) interactive table holding information about the communities; can also be used to set the visualization on the canvas; (8) canvas with the node-link representation of the network. (9) text areas for further information about the visualization achieved. Notice that nodes shown belong to six communities, each with the color and size denoted in the table (7). These elements are further described in this section.

views expressed do not reflect the official policy or position of FAPESP or CNPq.

References

1. Jolliffe I. *Principal component analysis*. Springer, 2011.

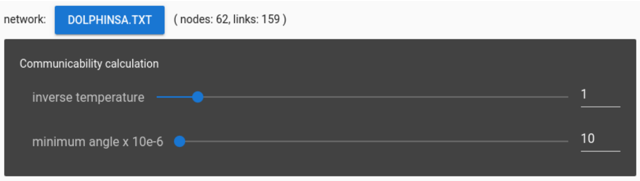

Figure 2. The initial step of the ComNetVis usage is to apply the communicability calculation to an input network. The button at the top is for network uploading and selection. In the gray box, the user chooses the inverse temperature and the minimum angle.

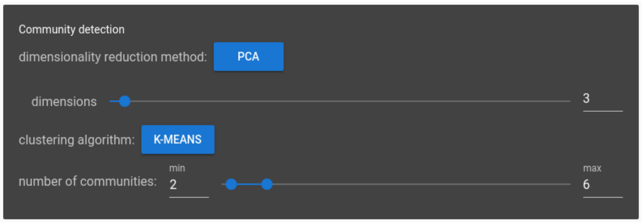

Figure 3. Settings of the community detection: the dimensionality reduction method; the target dimensionality in which the communities are going to be found; the clustering algorithm; and the minimum and maximum number of communities.

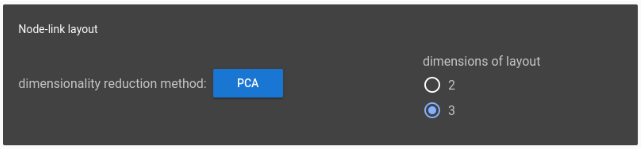

Figure 4. Settings of the layout: the dimensionality reduction method and the number of dimensions of the layout, which can be 2 or 3.

2. Maaten Lvd and Hinton G. Visualizing data using t-sne. *Journal of machine learning research* 2008; 9(Nov): 2579–2605.
3. Roweis ST and Saul LK. Nonlinear dimensionality reduction by locally linear embedding. *science* 2000; 290(5500): 2323–2326.
4. Tenenbaum JB, De Silva V and Langford JC. A global geometric framework for nonlinear dimensionality reduction. *science* 2000; 290(5500): 2319–2323.
5. Borg I and Groenen P. Modern multidimensional scaling: Theory and applications. *Journal of Educational Measurement* 2003; 40(3): 277–280.
6. Murtagh F and Legendre P. Ward's hierarchical clustering method: Clustering criterion and agglomerative algorithm. *arXiv preprint arXiv:11116285* 2011; .
7. Von Luxburg U. A tutorial on spectral clustering. *Statistics and computing* 2007; 17(4): 395–416.

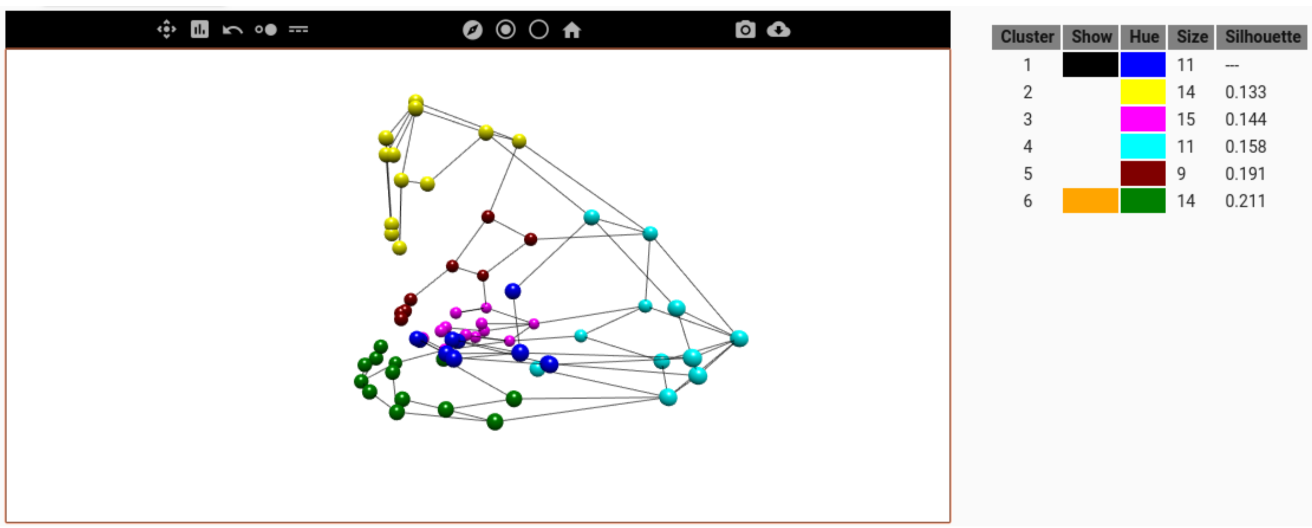

Figure 5. Second phase of ComNetVis usage: with the network is mapped to the canvas. The networks can be rotated or translated by click and drag or Ctrl-click and drag, respectively, in the canvas. The toolbar, above the canvas, holds three sets of buttons. The first set of buttons, which are at the left, enables: increase/decrease of node size; emphasize node size proportionality to degree; reset node proportionality; increase/decrease node transparency; and increase/decrease link transparency. The middle set of buttons, enables: show/hide centroid; show/hide best sphere center; show/hide best sphere surface; and recover initial position. The last set of buttons enables: save image; and download communities as a CSV file. The table, to the right, allows the analyst to change the number of communities associated to the network, and to change the color of the community, by clicking on the columns “Show” and “Hue”, respectively. The “Size” of each community changes in the table when the user changes the number of communities to be mapped to the screen. The “Silhouette” score is a measure of how consistent are the clusters: higher values are associated to better clusters.