Community Finding with Applications on Phylogenetic Networks

Luís Rita[1,3,4]

Supervision Team: Alexandre Francisco[1,2], João Carriço[3], Vítor Borges[4]

(1) Instituto Superior Técnico, Universidade de Lisboa, Lisboa, Portugal
(2) INESC-ID, R. Alves Redol 9, 1000-029 Lisboa, Portugal
(3) Instituto de Microbiologia, Instituto de Medicina Molecular, Faculdade de Medicina, Universidade de Lisboa, Lisboa, Portugal
(4) Departamento de Doenças Infeciosas, Instituto Nacional de Saúde Dr. Ricardo Jorge, Lisboa, Portugal

Abstract

With the advent of high-throughput sequencing methods, new ways of visualizing and analyzing increasingly amounts of data are needed. Although some software already exist, they do not scale well or require advanced skills to be useful in phylogenetics.

The aim of this thesis was to implement three community finding algorithms – Louvain, Infomap and Layered Label Propagation (LLP); to benchmark them using two synthetic networks – Girvan-Newman (GN) and Lancichinetti-Fortunato-Radicchi (LFR); to test them in real networks, particularly, in one derived from a *Staphylococcus aureus* MLST dataset; to compare visualization frameworks – Cytoscape.js and D3.js, and, finally, to make it all available online (mscthesis.herokuapp.com).

Louvain, Infomap and LLP were implemented in JavaScript. Unless otherwise stated, next conclusions are valid for GN and LFR. In terms of speed, Louvain outperformed all others. Considering accuracy, in networks with well-defined communities, Louvain was the most accurate. For higher mixing, LLP was the best. Contrarily to weakly mixed, it is advantageous to increase the resolution parameter in highly mixed GN. In LFR, higher resolution decreases the accuracy of detection, independently of the mixing parameter. The increase of the average node degree enhanced partitioning accuracy and suggested detection by chance was minimized. It is computationally more intensive to generate GN with higher mixing or average degree, using the algorithm developed in the thesis or the LFR implementation. In *S. aureus* network, Louvain was the fastest and the most accurate in detecting the clusters of seven groups of strains directly evolved from the common ancestor.

Community Finding | Phylogenetic Networks | Data Visualization | Web Application

1. Introduction

According to *The Economist*, data has become the most important resource in the world [1]. New ways to storage, analyze and visualize increasing amounts of information are becoming a pressing need.

Interactions among different elements of real systems can be seen as networks [2]. From their topology or pattern of connections, functional knowledge can be extracted. Although empirical analysis can be useful in small networks, for big data it is indispensable the use of algorithms to uncover their properties. Consequently, mathematical models called graphs are used to simplify their representation. Graph theory provides all the tools available to analyze them. This representation implies all the elements of the network and their relations must be acquired. Generally, the closer to the human population is the analysis, the more difficult it gets to gather data. The two main reasons are the lack of non-invasive acquisition methods or privacy issues [3]. In fact, in health, legislation is becoming progressively stricter in respect to patient's data protection [4]. On one hand, it is fundamental to assure ethical use of medical data, on the other, it may constitute a hurdle for investigating the causes of many pathologies. In the next lines, the discussion will focus in the specific topic of infectious diseases.

As stated by the *World Health Organization*, Antimicrobial resistance (AMR) is in the top ten of major threats demanding attention in 2019 [5]. In the last years, AMR has been potentiated by the use of drugs intended to treat diseases caused by pathogenic organisms. Depending on the organism, it is subdivided in antiviral (virus), antibiotic (bacteria), antifungal (fungi) or antiparasitic (parasites) resistance. Although resistance is a natural evolutionary process, in which the most competent strains get selected over time, the frequency of use, or even misuse, of antimicrobial drugs is accelerating the growth of resistance rates for several microbial species [6]. This is particularly relevant not only among the human population, but also in farms where cattle are being given preventive doses of antibiotics as growth adjuvants. Moreover, the frequency people are travelling around the world is facilitating the spread of resistant organisms. Due to all of this and the high mortality of

infectious diseases (Figure 1), it is fundamental to empower central health authorities with scalable and easy-to-use tools to allow their intervention in short time scales [7].

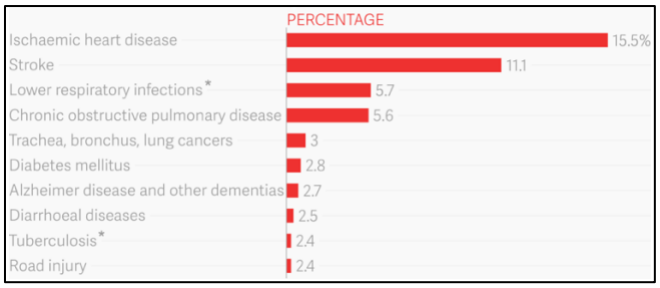

Figure 1 Infections are still one of the major causes of death (*). Data collected by WHO between 2000-2015 from a population of over 90 000 people, in WHO member states. Adapted from [8].

Next Generation Sequencing methods, developed in the last years, have been providing us massive quantities of microbiological data [9]. In fact, due to the high discriminative power of whole genome sequencing methods, it has been possible to infer more precise evolutionary relationships among strains. Consequently, allelic profiles with higher number of genes keep increasing and, with them, more complete phylogenetic trees or networks depicting evolutionary distances among strains can be assessed. All of this has been useful in the multiple stages of disease prevention: vaccine design, assess the pathogenicity of a sequenced strain, detect outbreaks and in infection control. As well as diagnosis, enhancing the detection of mixed infections (caused by multiple genetically distinct organisms) [10].

Generally, the goal of this thesis was to extract functional knowledge from the topological structure of phylogenetic networks. For instance, to be able to infer properties like resistance to antimicrobials, based on the similarity and characteristics of the other organisms in the network. Contrarily to random networks, real ones present an asymmetrical distribution of edges. From this asymmetry, clusters of nodes can be identified. By definition, communities are regions in the network with higher density of edges. Nonetheless, this definition is not restrictive enough to undoubtedly identify them in graphs. This raises the following question: how can we detect communities if it is not clear what we are looking for? To answer this question, community finding theory has been developed in the past years. In spite there has been a considerable advance in terms of accuracy and speed of detection, some algorithms perform better than others in different networks.

1.1. Objectives

The aim of the thesis was to implement three community finding algorithms in JavaScript – Louvain, Infomap and Layered Label Propagation (LLP); to benchmark them, in terms of accuracy and speed, against two synthetic networks – Girvan-Newman (GN) and Lancichinetti-Fortunato-Radicchi (LFR) networks; to perform additional testing using an Amazon, Zachary's Karate Club and *Staphylococcus aureus* Multilocus Sequence Typing (MLST) Single Locus Variant (SLV) network; to compare different visualization frameworks in terms of their tools and robustness to big data - D3.js (using SVG and Canvas elements) and Cytoscape.js, and to create a web application to allow network visualization (before and after running community finding algorithms) and plot all data obtained from the benchmark tests.

1.2. Structure Outline

In Methodology, the implemented community finding algorithms, implemented benchmark tools, real test data and data visualization libraries are introduced. In Results, community finding algorithms are analyzed in terms of accuracy and speed. The speed of generation of benchmark networks is determined based on the different properties of the networks. Different web visualization frameworks are compared. And a *S. aureus* MLST SLV network is partitioned using different algorithms and input parameters. In Conclusion, a retrospective analysis on whether the goals were attained or not is made, main difficulties, future work to be performed in INSaFLU web application is detailed and a set of prospective improvements to the work already done is suggested.

2. Methodology

2.1. Community Finding

Louvain Algorithm

This algorithm, which was implemented in the thesis, is divided in 2 phases: Modularity Optimization and Community Aggregation [11]. Together, they are considered 1 pass. After the first step is completed, the second follows. Both are executed until there are no more changes in the network and maximum modularity is achieved.

Modularity Optimization – the algorithm will randomly order all nodes in the network such that, one by one, it will remove and insert them in a different community C. This will continue until no significant increase of modularity (input parameter) is verified:

$$\Delta M = \left[\frac{\Sigma_{in} + k_{i,in}}{2m} - \left(\frac{\Sigma_{tot} + k_i}{2m}\right)^2\right] - \left[\frac{\Sigma_{in}}{2m} - \left(\frac{\Sigma_{tot}}{2m}\right)^2 - \left(\frac{k_i}{2m}\right)^2\right] \quad (1)$$

Let Σ_{in} be the sum of the weights of the links inside C, Σ_{tot} the sum of the weights of all links to nodes in C, k_i sum of the weights of all links incident in node i, $k_{i,in}$ the sum of the weights of links connecting node i and nodes in the community C and m is the sum of the weights of all edges in the graph.

Community Aggregation – After finishing the first step, all nodes belonging to the same community are merged into a single giant node. Links connecting giant nodes are the sum of the ones previously connecting nodes from the same different communities. This step also generates self-loops which are the sum of all links inside a given community before being collapsed into one node.

Infomap Algorithm

Similarly to Louvain, Infomap algorithm was also implemented. It partitions the network using a different quality function – minimum description length [12]:

$$L(M) = \left(\sum_{m\in M} q_m\right) log\left(\sum_{m\in M} q_m\right) - 2\sum_{m\in M} q_m\, log(q_m) - \sum_{\alpha\in V} p_\alpha\, log(p_\alpha) + \sum_{m\in M}(q_m + \sum_{\alpha\in m} p_\alpha)\, log\left(q_m + \sum_{\alpha\in m} p_\alpha\right) \quad (2)$$

Let q_m be the exit probability of a module m and p_α the relative weight w_α that is computed dividing the total weight of the edges connected to α by twice the total weight of all links in the graph.

Layered Label Propagation Algorithm

LLP algorithm development was based in Label Propagation (LP) [13]. The latter starts by assigning a different community to each node in the network and then, iteratively, it modifies the respective community based on the dominant one from the nodes in the immediate neighborhood. In each iterate, this step is executed for all nodes, in a previously randomized order.

LLP not only takes into account the nodes in the neighborhood, but also the ones in the remaining network [14]. The iterative process is the same, the difference resides in the value it optimizes. In this case, the assigned community is the one that maximizes:

$$k_i - \gamma(v_i - k_i) \quad (3)$$

Being k_i the number of nodes with label λ_i in the neighborhood of a given node and v_i the total number in the whole graph labeled the same way.

This approach, which was also implemented, buffers the number of nodes belonging to a certain community (in the node's neighborhood), by its usual presence across the whole network.

Table 1 Time complexity of community finding algorithms.

Algorithm	Time Complexity	Reference
Louvain	$O(L)$	[11]
Infomap	$O(NlogN)$	[12]
LP	$O(L)$	[13, 15]
LLP	$O(L)$	[14]

2.2. Benchmark

Different community finding algorithms partition the same graph in different ways. Consequently, it is important to understand which algorithms are the most indicated.

An accurate way to gain this insight is by partitioning networks whose community structure is known. GN and LFR benchmark networks are presented in the next sub-sections, as well as NMI which estimates the similarity between the original and the detected partitions.

Community finding algorithms were not only tested in terms of accuracy, but also speed. All benchmark tests were repeated 10 times. The mean values and corresponding confidence intervals at 95% were included in Results.

Girvan-Newman Network

In GN network, 128 nodes are divided in 4 groups, each one with exactly 32 nodes [16]. Then, a mixing parameter, chosen by the user, defines the probability of nodes from different communities being connected. This probability is given by:

$$\mu = \frac{k^{ext}}{k^{int} + k^{ext}} \quad (4)$$

Each node is connected to 16 others (Figure 2). The algorithm hereby implemented not only allows the mixing parameter to vary, but also the average degree. This led to an increased number of benchmark tests using these networks.

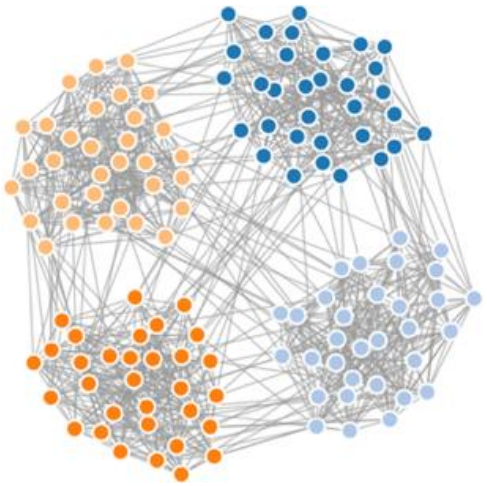

Figure 2 GN synthetic network. $N = 128$, $\mu = 0.1$ and $k = 16$. Represented using D3.js and SVG.

Lancichinetti-Fortunato-Radicchi Network

LFR benchmark networks (Figure 3) [17, 16] try to better approximate the real ones. This means that not only they

consider a power-law distribution of community sizes with coefficient ζ (5), but also a power-law distribution of node degrees with coefficient γ (6):

$$p_{N_c} \sim N_c^{-\zeta} \quad (5)$$

$$p_k \sim k^{-\gamma} \quad (6)$$

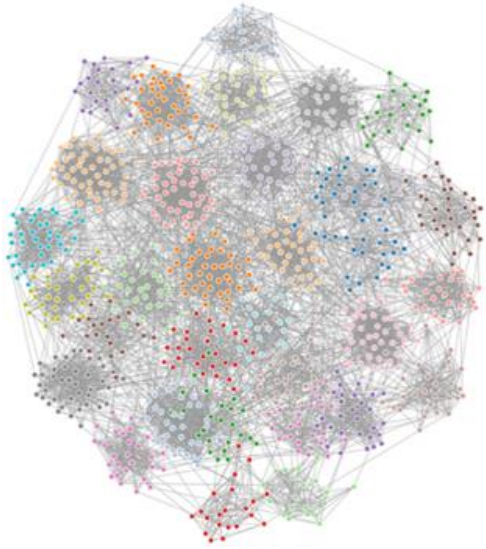

Figure 3 LFR synthetic network. $N = 1000$, $\mu = 0.1$, $k_{avg} = 15$, $k_{max} = 50$, $c_{min} = 20$ and $c_{max} = 50$. Represented using D3.js and SVG.

Normalized Mutual Information

The clustering quality of the community finding algorithms was tested using Normalized Mutual Information (NMI) [18]. An algorithm was developed, in JavaScript, in the thesis and tested in several networks. It receives an input of 2 arrays, with the same length, and returns the corresponding NMI:

$$NMI(Y, C) = \frac{2 \times I(Y; C)}{[H(Y) + H(C)]} \quad (7)$$

$$I(Y; C) = H(Y) - H(Y|C) \quad (8)$$

NMI is dependent on the mutual information I, the conditional entropy $H(Y|C)$ and the entropy of the labeled $H(Y)$ and clustered set $H(C)$.

2.3. Real Test Data

The higher the number of tests performed using the implemented algorithms, the higher the certainty they are working properly for the broadest number of networks. This way, disconnected networks (sampled from an Amazon network), small sized ones (Zachary's Karate Club) and a phylogenetic network (*S. aureus* MLST SLV) were additionally considered. Nevertheless, no benchmark tests were performed using those.

Amazon Network

This network was obtained after crawling Amazon website [19]. Each node represents a given product available in the store and a connected pair means they are frequently bought together. The original network contains 334 863 nodes and 925 872 links. Only the first 5 000 links along the respective nodes were used in the tests (Figure 4).

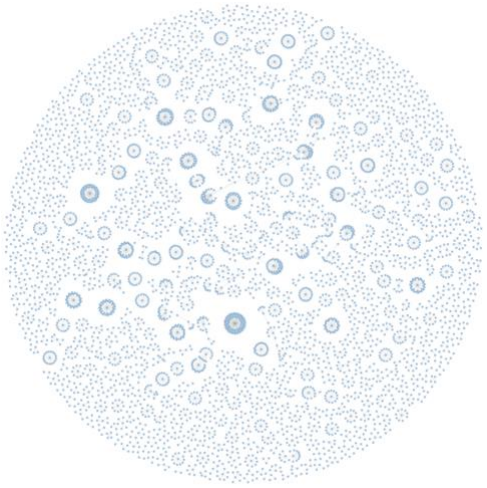

Figure 4 Amazon product co-purchasing network sampled from a 5 000 links graph. Represented using D3.js and SVG.

Zachary's Karate Club Network

This is a social network representing 77 relations between 34 individuals from a university karate club [20].

Node 1 is the instructor and 34 is the president (Figure 5). Dark and light blue nodes represent the split inside the club that was created due to a conflict between them. Each link connects 2 members when they used to meet regularly outside the club, before splitting.

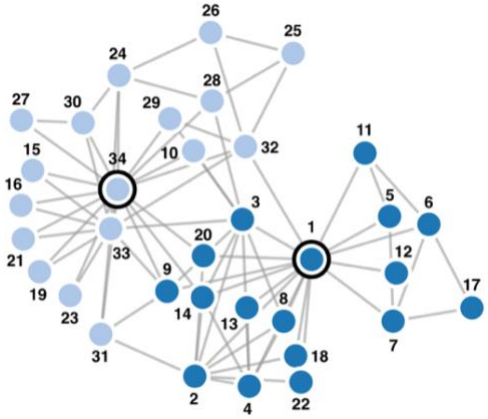

Figure 5 Zachary's karate club network. Represented using D3.js and SVG.

Staphylococcus aureus

To test the application of the community finding algorithms in a phylogenetic network, an MLST database of allelic profiles for *S. aureus* was used (Figure 6) [21]. The MLST profiles contain information about seven *loci* (*arcC*, *aroE*, *glpF*, *gmk*, *pta*, *tpi* and *yqiL*) and, at the time of download, it had 5199 MLST profiles (ST). A graph was created by linking each ST to all of its SLVs and the largest component was used for the benchmark of the community finding algorithms. The choice of this test network was based on the fact it can emulate similar patterns to the ones that are created using cgMLST or wgMLST profiles, which have many more loci than MLST.

Figure 6 *S. aureus* MLST SLV network. Represented using D3.js and Canvas.

2.4. Visualization Frameworks

In the visualization application, the user can opt by visualizing every network using D3 or Cytoscape JavaScript libraries. In the case of the first framework, there is the possibility of visualizing data through a Canvas or an SVG element. All networks were plotted using a force directed implementation. This mode requires the developer to set features like the gravity, repulsion or spring constant which will affect how the network is displayed. Depending those parameters, node and link overlapping should be reduced in order to get an enhanced view of the graph (Figure 7).

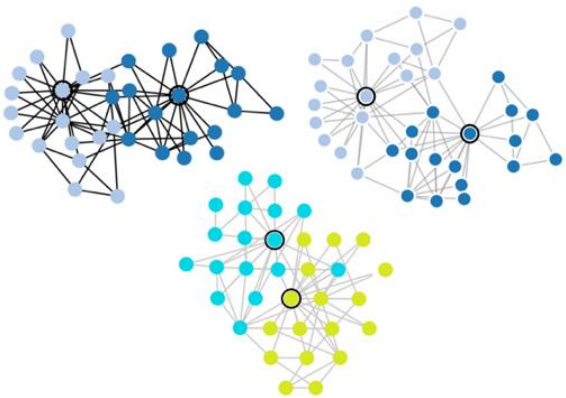

Figure 7 Zachary's Karate Club network represented using D3.js Canvas (top-left), D3.js SVG (top-right) and Cytoscape.js (bottom).

Regarding benchmark plotting, D3.js was used. Its flexibility in terms of visual representation, performance and the number of practical examples available on the web, determined this choice.

3. Results

3.1. Web Application

In order to facilitate the analysis and visualization of the results after the execution of the previous algorithms, a web application was implemented (mscthesis.herokuapp.com).

Its backend was conceived using Node.js and it runs in a Heroku server which is linked to a GitHub repository (github.com/warcraft12321/Thesis) containing all the implementations and documents related to the thesis. A Digital Object Identifier (DOI) was attributed to this repository using Zenodo (zenodo.org/badge/latestdoi/162063699). An image of the app is available at Docker Hub (cloud.docker.com/u/warcraft12321/repository/docker/warcraft12321/thesis). The following actions are strictly executed in the cloud:

- Generation of GN and LFR benchmark networks. The latter are generated using a binary from a C++ implementation [22]. This was possible after importing a package from NPM which simulates the command line in the server;
- The real test networks had their data properly processed in the cloud, so that it could be sent and displayed in the frontend;
- Once the execution of Louvain, Infomap and LLP algorithms is not performed in the user's browser, this may raise some privacy issues. Although, it was not feasible to run Infomap in the user side and still keep the web application functional in all devices;
- The communication between the app and NPM was established in the server, so that it is possible to import and plot the statistics from each package uploaded to NPM in the application;
- Finally, all benchmark data was obtained after running the algorithms in the cloud.

The frontend is both static and dynamic. The static counterpart uses HTML, CSS and JavaScript. The dynamic execution is managed by Node.js, which runs in the server-side.

3.2. Community Finding Algorithms

In terms of accuracy, Louvain outperformed all others for weakly mixed networks ($\mu < 0.4$). It is followed by Infomap, LP and LLP ($\gamma = 0.5$) for $\mu < 0.3$ or Infomap, LLP and LP ($\gamma = 0.5$) for $0.3 \leq \mu \leq 0.4$. By increasing μ, LLP, which is able to detect communities based on the panorama of the whole network, consistently becomes the most effective for $\mu \geq 0.5$. Comparing Louvain and Infomap when $\mu \geq 0.5$, their performance is similar. Based on the CI 95%, it is not possible to state one performs better than the other. Another point to highlight is that Louvain and Infomap are very sensitive, consequently, unreliable detecting communities in networks with $\mu \approx 0.5$. LLP and LP do not present any specific value in which community detection is steeply affected. These conclusions are all valid to GN and LFR networks (Figure 8 and Figure 9).

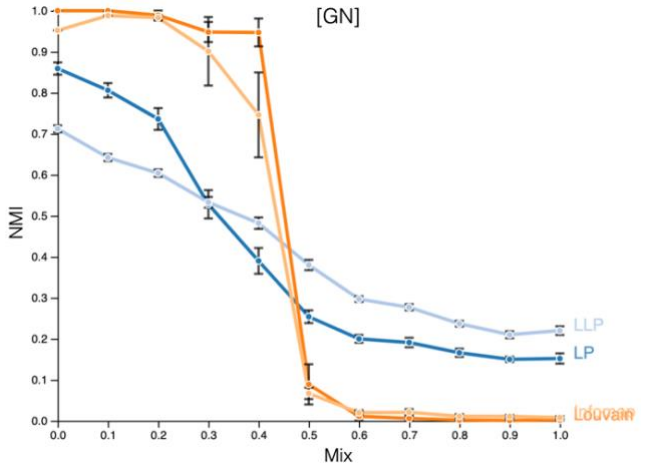

Figure 8 Clustering quality of each algorithm: Louvain, Infomap, LP and LLP in terms of the mixing parameter in GN networks, with the remaining properties as in Figure 2.

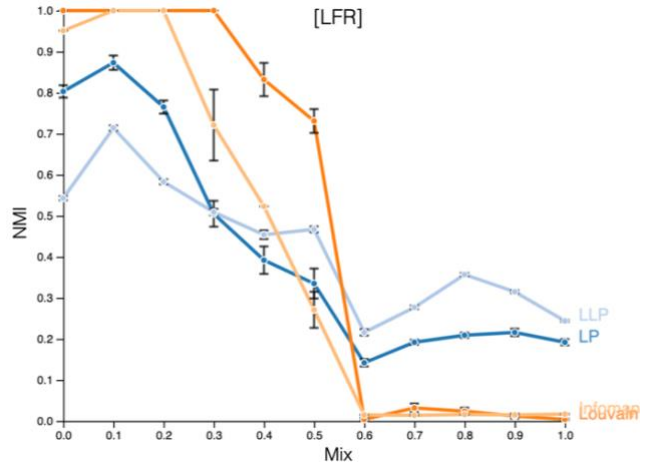

Figure 9 Clustering quality of each algorithm: Louvain, Infomap, LP and LLP in terms of the mixing parameter in LFR networks, with the remaining properties as in Figure 3.

Having considered only two gamma values in LLP, a more precise analysis was needed to check whether there could be others achieving better results.

In the case of GN network (Figure 10), it is possible to detect communities with higher accuracy in networks less mixed. The performance decreases whenever we increase γ for $\mu < 0.5$. When $\mu \geq 0.5$, the NMI between the detected partition and the one in the original network is higher as long as we keep increasing γ. Another point to highlight is the possibility of detecting communities with higher precision in networks with higher μ using appropriate γ than in others with lower μ but with a non-optimal γ. This is valid for networks with $\mu > 0.5$ (Figure 11).

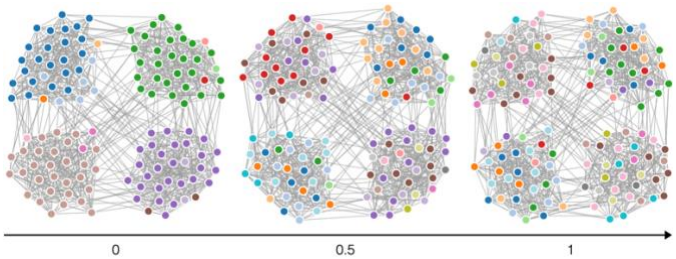

Figure 10 GN network (Figure 2) after running LLP. When γ is closer to 0, the coarse structure of the network with few, big and sparse communities is highlighted. As γ grows, fine-grained structure is unveiled. Communities become smaller and denser. Represented using D3.js and SVG.

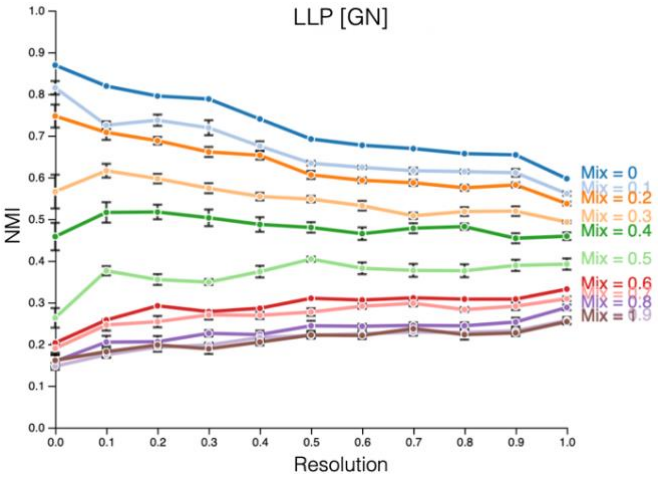

Figure 11 LLP algorithm accuracy in terms of the resolution parameter. Analysis performed for mixing parameters 0 – 1 (blue-brown) in GN network, with the remaining properties as in Figure 2.

In LFR networks, increasing gamma always leads to a decrease in the NMI (Figure 12).

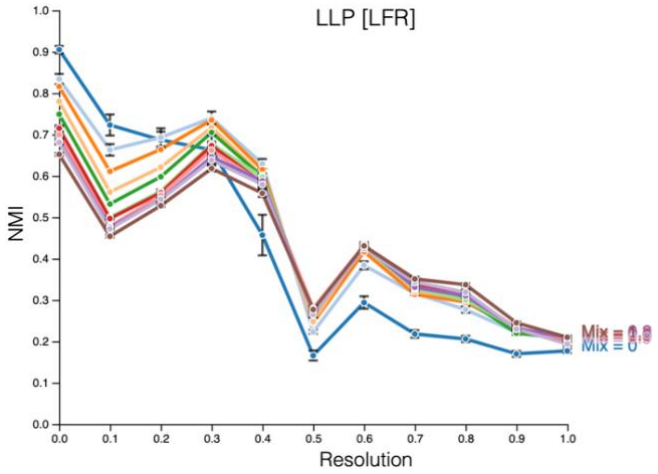

Figure 12 LLP algorithm accuracy in terms of the resolution parameter. Analysis performed for mixing parameters 0 – 1 (blue-brown) in LFR network, with the remaining properties as in Figure 3.

The influence of the average node degree was tested to check whether it affects the capacity of each algorithm to identify communities.

In the case of GN networks, for $\mu \leq 0.5$, an increased average degree enhances the capacity of Louvain, Infomap, LP and LLP to differentiate communities. For $\mu > 0.5$, NMI clustering quality is lower as higher is the average node degree. Assuming it is not expected the algorithms to identify the original communities for $\mu \approx 1$, this suggests that higher average degrees tends to decrease detection by chance. This observation can be applied to GN and LFR networks (Figure 13).

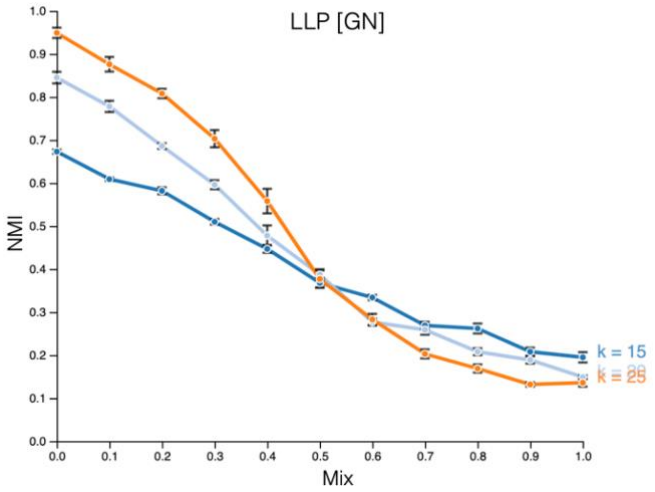

Figure 13 LLP algorithm accuracy in terms of the mixing parameter. GN network with average node degrees: 15, 20 and 25. The remaining properties are similar to the ones in Figure 2.

In terms of speed, Louvain algorithm performed significantly better than any other. One important property that contributes to its efficiency is that it only calculates modularity variation at each iteration of Modularity Optimization phase (1). In Infomap, minimum description length is recalculated at the end of each iteration of the optimization step (2). Due to this computationally intensive step, Infomap presented the poorest performance in the benchmark tests. Based on Table 1, it was expected the difference of execution time between Infomap and all others to be logarithmic. Due to inefficient steps in the implementation, this was not achieved. Comparing LLP and LP, the fastest was the one that needed to optimize the simplest equation – LP (Figure 14). Although, time complexity appears to be the same, as predicted in Table 1. Again, using the same table, it was expected LLP and LP to execute about as fast as Louvain. The small difference observed in Figure 14 might be explained by some non-optimized step in the implementation of LLP/LP.

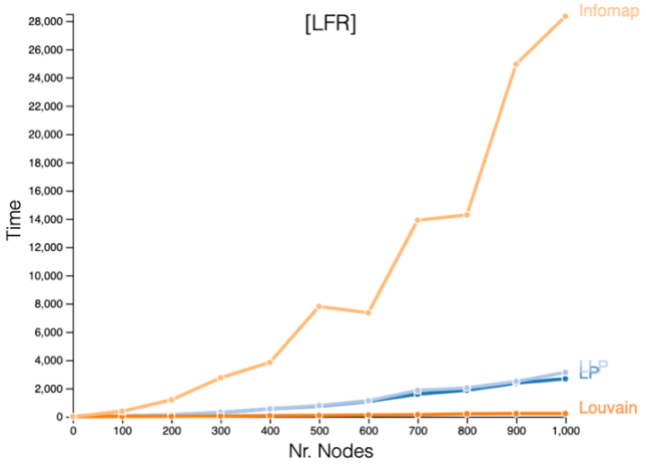

Figure 14 Time that Louvain, Infomap, LLP and LP took to finalize the analysis in terms of the size of the LFR networks. The remaining properties are similar to the ones in Figure 3.

3.3. Benchmark Networks Algorithms

Previous GN and LFR networks were generated so the previous tests could be performed. The algorithm to create the former was implemented in the thesis. The time it takes to generate such networks increases exponentially with the average node degree and the mixing parameter (Figure 16). Becoming unfeasible to execute it for average degrees higher than 20, in an ordinary computer. Its performance was compared to the Fortunato's implementation which is highly scalable and allows the generation of networks with much higher number of links in shorter times (Figure 15). It was verified an approximately linear increase in the time of generation of networks with progressively higher mixing parameters, contrarily to the one implemented in the thesis. Just like before, generating networks with more edges, requires additional computational power.

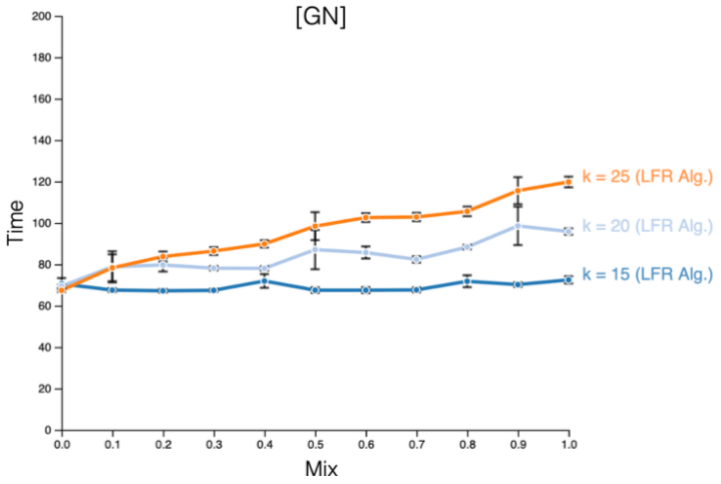

Figure 15 Time needed to generate GN networks (using LFR implementation) in terms of their mixing parameter. Analysis performed in networks with average node degrees: 15, 20 and 25. The remaining properties are similar to the ones in Figure 2.

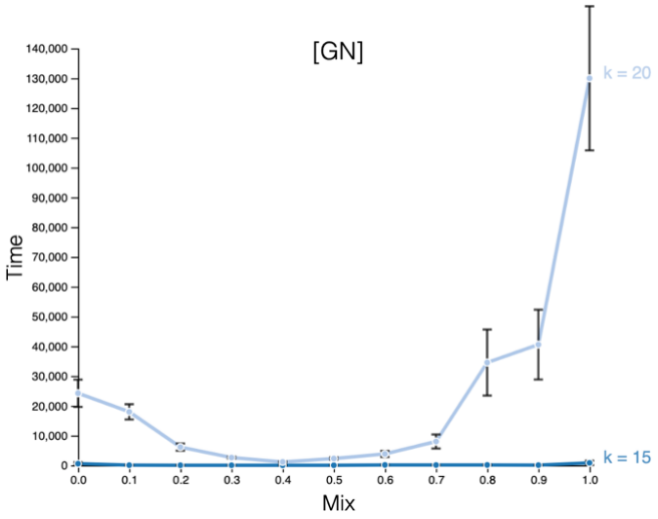

Figure 16 Time needed to generate GN networks (using thesis implementation) in terms of their mixing parameter. Analysis performed in networks with average node degrees: 15 and 20. The remaining properties are similar to the ones in Figure 2.

Additional tests were performed in LFR networks. The execution time was tested against the number of nodes and

the mixing parameter of the network. In both cases, it was verified an approximately linear relation (Figure 17 and Figure 18). The number of nodes was not considered in GN networks, once they have a fixed number of elements per community and of communities. In spite only the mixing parameter and the number of nodes were analyzed, the user can still choose to adjust nodes maximum degree, the exponents of the nodes degree distribution and communities size distribution, the maximum/minimum number of nodes per community, the number of overlapping nodes per community (covers were not considered in the thesis) and the number of memberships for the overlapping nodes [22].

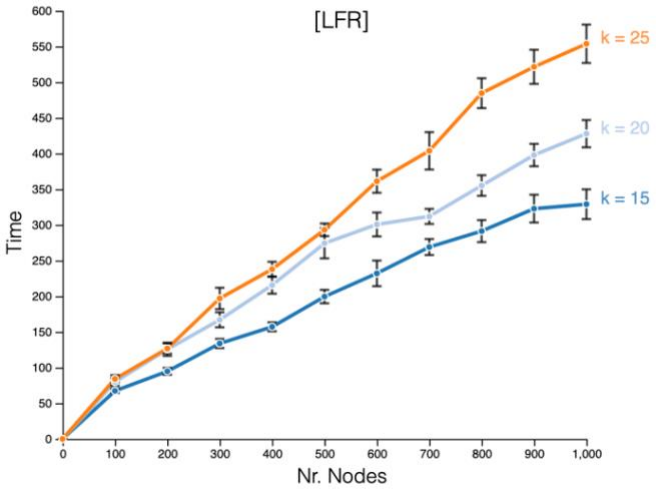

Figure 17 Time needed to generate LFR networks in terms of their number of nodes. Analysis performed in networks with average node degrees: 15, 20 and 25. The remaining properties are similar to the ones in Figure 3.

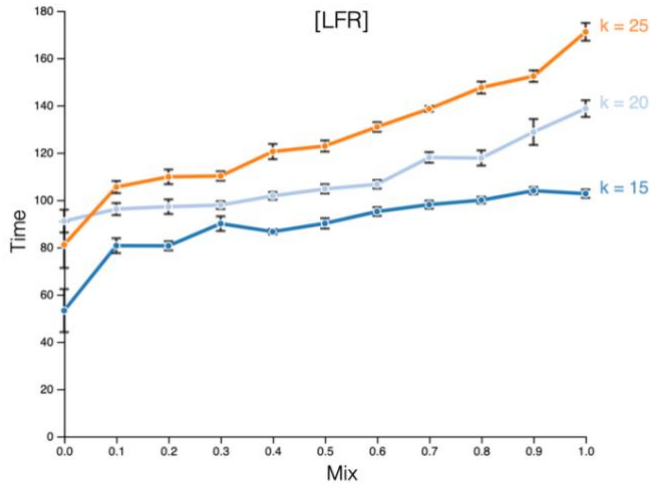

Figure 18 Time needed to generate LFR networks in terms of their mixing parameter. Analysis performed in networks with average node degrees: 15, 20 and 25. The remaining properties are similar to the ones in Figure 3.

Every community finding, benchmark network generator, accuracy measure and related algorithms were made available in NPM (Table 2). This way, they can be used in any user-specific application.

Table 2 Algorithms implemented in the thesis were made available in NPM.

Algorithm	Package Name	URL
Louvain	*louvain-algorithm*	npmjs.com/package/louvain-algorithm
Infomap	*infomap*	npmjs.com/package/infomap
LLP	*layered-label-propagation*	npmjs.com/package/layered-label-propagation
GN Network Generator	*girvan-newman-benchmark*	npmjs.com/package/girvan-newman-benchmark
NMI	*normalized-mutual-information*	npmjs.com/package/normalized-mutual-information
Hamming Distance	*hamming-dist*	npmjs.com/package/hamming-dist

3.4. Visualization Frameworks

D3.js (SVG) allowed the user to visualize all networks used in the web application, as well as to represent the benchmark plots with the highest resolution possible (limited by density of pixels in the user's device). In terms of performance, it is the second fastest.

D3.js (Canvas) is the fastest among the analyzed frameworks. Nevertheless, when compared to the previous, the nodes and links from the network lose resolution.

Cytoscape.js is advisable when network analysis will be performed along with its representation. There is a considerable number of algorithms, including community finding ones, that can facilitate the study of the graph. The lack of working examples and the slow processing of network data made Cytoscape.js the less efficient. Representation of *S. aureus* SLV network and others with similar or higher number of nodes/edges overflows the web application.

3.5. *Staphylococcus aureus*

Louvain, Infomap and LLP algorithms were executed in the biggest component of the SLV graph obtained from the MLST dataset of *S. aureus*.

Louvain algorithm was executed using three modularity variation thresholds: 0.1, 0.02 and 0.01. It is possible to verify that high modularity partitions are rapidly achieved (2 passes needed to obtain final result), as predicted in Introduction. This characteristic makes modularity-based algorithms fast.

Using Infomap algorithm, several minimum description length thresholds were considered: 0.1, 0.02 and 0.01. For all values, every small dense group of nodes descending from the same ancestor was separated in different communities. Thus, it was not further considered.

LLP algorithm considered three values for the resolution parameter: 0, 0.5 and 1. Similarly to Infomap, it partitioned the network in several small and dense communities.

From the benchmark analyses of Louvain, Infomap and LLP against GN network (Figure 8 and Figure 9), it was predictable Louvain would perform better than the others, once it run in a network with well-defined communities. After adjusting the stopping parameter of Louvain, such that the communities identified were the most pertinent – nodes from densely connected regions descendent from the same ancestor belonging to the same community, a final partition was obtained (Figure 19). It is important to note that some clusters analogous to the one highlighted with a square were subdivided by the algorithm. One explanation is the presence of high recombination rates between strains of different sub-groups.

By uploading to PHYLOViZ Online the generated metadata file with the nodes labeled accordingly to the inferred communities, one may observe the original phylogenetic tree with the same communities identified.

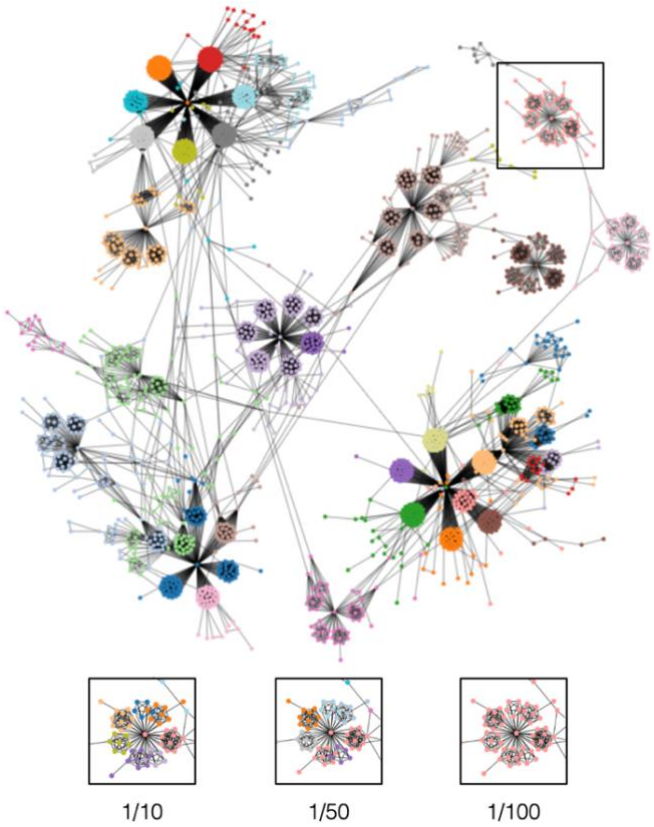

Figure 19 Partition obtained after running Louvain algorithm (considering different thresholds for the modularity variation) on Clonal Complex 0 (as identified in PHYLOViZ 2) of *S. aureus* MLST SLV network. Represented using D3.js and Canvas.

4. Conclusion

Louvain, Infomap and LLP algorithms were implemented in JavaScript. Unless otherwise stated, next conclusions are valid for GN and LFR networks. In terms of speed, Louvain outperformed all others. In terms of accuracy, in networks with well-defined communities, Louvain was the most accurate. For higher mixing, LLP was the best. Contrarily to weakly mixed, it is advantageous to increase the resolution parameter in highly mixed GN networks. In LFR networks, higher resolution decreases the accuracy of detection, independently of the mixing parameter. The increase of the average node degree enhanced partitioning accuracy, suggesting detection by chance was minimized. It is computationally more intensive to generate GN networks with higher mixing or average degree, using the algorithm developed in the thesis or the LFR implementation. In the *S. aureus* MLST SLV network, Louvain was the fastest and the most accurate in detecting the clusters of 7 groups of strains evolved from the same ancestor. The lower the modularity variation threshold, the better the detection.

Along with the three community finding algorithms, GN benchmark network generator, NMI and hamming distance algorithms were made available in NPM. The web application is hosted in the Heroku cloud platform. An image containing all the required modules to run the app in a local machine can be download from Docker Hub. All the implementations and a complete roadmap of the thesis was made available in GitHub. A Digital Object Identifier (DOI) was attributed to this repository using Zenodo.

One of the main difficulties encountered was running community finding algorithms in the user-side. A feature that would rise less privacy concerns. It was necessary to run the algorithms in the server, so that the application would not crash. It was not possible to reach, in the Infomap implementation, the computational efficiency predicted by its time complexity. The online visualization of networks with more than 10 000 nodes was unfeasible, even using the framework that showed the best performance – D3.js (using Canvas element).

During the next months, some of the tools developed in this thesis will be implemented in INSaFLU. New visualization frameworks for displaying phylogenetic trees, along with the uploaded metadata, will enhance the traceability of the evolutionary path among influenza strains. A geographical map dynamically varying with time will allow the user to precise the moment and location at which each sample was obtained and enhance flu surveillance during the critical season. An interactive histogram displaying each nucleotide belonging to a consensus sequence obtained from a sample of influenza virus with the SNPs highlighted, as well as with their effects in the phenotype of the strain specified. New flu surveillance dynamic reports will allow the visualization of increased quantities of data in a minimalistic way.

As future improvements, the implemented algorithms can be optimized in terms of memory and speed. Network and chart visualization may run faster in a more efficient implementation. The previous updates will make the web application to load quicker and run smoother. To benchmark community finding algorithms with a more representative set of networks (spanning a wider range of topological

properties) would make possible to infer, with a higher certainty, if any of the community finding algorithms consistently performs better in phylogenetic networks. The integration of community finding functionalities developed in the thesis in a phylogenetic/surveillance-oriented analysis web application like PHYLOViZ Online or INSaFLU would be beneficial.

References

[1] "The world's most valuable resource is no longer oil, but data," The Economist, 6 May 2017. [Online]. Available: https://www.economist.com/leaders/2017/05/06/the-worlds-most-valuable-resource-is-no-longer-oil-but-data. [Accessed 28 May 2019].

[2] A.-L. Barabási, "Network Science Book," [Online]. Available: http://networksciencebook.com. [Accessed 15 May 2019].

[3] H. G. S. Patil, A. N. Babu and P. S. Ramkumar, "Non-invasive data acquisition and measurement in bio-medical technology: An overview," in *Maximizing Healthcare Delivery and Management through Technology Integration*, IGI Global, 2016.

[4] "Health," EUROPEAN DATA PROTECTION SUPERVISOR, [Online]. Available: https://edps.europa.eu/data-protection/our-work/subjects/health_en. [Accessed 9 June 2019].

[5] "Ten threats to global health in 2019," World Health Organization, [Online]. Available: https://www.who.int/emergencies/ten-threats-to-global-health-in-2019. [Accessed 4 June 2019].

[6] "Antimicrobial resistance," World Health Organization, 15 February 2018. [Online]. Available: https://www.who.int/en/news-room/fact-sheets/detail/antimicrobial-resistance. [Accessed 29 May 2019].

[7] Z. A. Memish, S. Venkatesh and A. M. Shibl, "Impact of travel on international spread of antimicrobial resistance," *International Journal of Antimicrobial Agents,* vol. 21, no. 2, pp. 135-142, 2003.

[8] "Top 10 Leading Causes of Death Globally," [Online]. Available: https://www.theatlas.com/charts/HkLaDreuW. [Accessed 12 May 2019].

[9] B. Ribeiro-Gonçalves, A. P. Francisco, C. Vaz, M. Ramirez and J. A. Carriço, "PHYLOViZ Online: web-based tool for visualization, phylogenetic inference, analysis and sharing of minimum spanning trees," *Nucleic Acids Research,* vol. 44, no. 1, pp. 246-251, 2016.

[10] Y. Motro and J. Moran-Gilad, "Next-generation sequencing applications in clinical bacteriology," *Biomolecular Detection and Quantification,* vol. 14, p. 1–6, 2017.

[11] V. D. Blondel, J.-L. Guillaume, R. Lambiotte and E. Lefebvre, "Fast unfolding of communities in large networks," *J. Stat. Mech. (2008) P10008,* p. 12, 2008.

[12] M. Rosvall, D. Axelsson and C. T. Bergstrom, "The map equation," *The European Physical Journal Special Topics,* vol. 178, no. 1, pp. 13-23, 2009.

[13] N. Raghavan, R. Albert and S. Kumara, "Near linear time algorithm to detect community structures in large-scale networks," *Physical Review E 25th Anniversary Milestones,* vol. 76, no. 3, 2007.

[14] P. Boldi, M. Rosa, M. Santini and S. Vigna, "Layered label propagation: a multiresolution coordinate-free ordering for compressing social networks," in *WWW '11 Proceedings of the 20th international conference on World wide web*, 2011.

[15] L. Šubelj, "Label propagation for clustering," in *Advances in Network Clustering and Blockmodeling*, New York, Wiley, 2018.

[16] M. Girvan and M. E. J. Newman, "Community structure in social and biological networks," *Proceedings of the National Academy of Sciences of the United States of America,* vol. 99, no. 12, pp. 7821-7826, 2002.

[17] A. Lancichinetti, S. Fortunato and F. Radicchi, "Benchmark graphs for testing community detection algorithms," *Physical review. E, Statistical, nonlinear, and soft matter physics.,* vol. 78, no. 4, 2008.

[18] A. Lancichinetti, S. Fortunato and J. Kertesz, "Detecting the overlapping and hierarchical community structure of complex networks," *New Journal of Physics,* vol. 11, 2009.

[19] J. Yang and J. Leskovec, "Defining and Evaluating Network Communities based on Ground-truth," in *Proceedings of 2012 IEEE International Conference on Data Mining (ICDM)*, 2012.

[20] W. Zachary, "An Information Flow Model for Conflict and Fission in Small Groups," *Journal of anthropological research,* vol. 33, 1976.

[21] "Staphylococcus aureus MLST Databases," PubMLST, 5 June 2019. [Online]. Available: https://pubmlst.org/saureus/. [Accessed 5 June 2019].

[22] A. Lancichinetti and S. Fortunato, "Benchmarks for testing community detection algorithms on directed and weighted graphs with overlapping communities," *Physical review. E, Statistical, nonlinear, and soft matter physics.,* vol. 80, 2009.