

# Multiscale Community Mining in Networks Using the Graph Wavelet Transform of Random Vectors

Nicolas Tremblay, Pierre Borgnat  
Physics Laboratory, CNRS UMR 5672, ENS Lyon  
University of Lyon, France  
firstname.lastname@ens-lyon.fr

**Abstract**—In an effort to simplify the analysis of data represented by networks, a classical approach is to uncover the community structure of the underlying graph. In this work, we take advantage of graph wavelets and the associated natural definition of scale to propose a multi-scale community mining tool. More precisely, at a given scale, we cluster nodes in the same community when their corresponding wavelets are highly correlated. We show that the wavelet transform of a few random signals is sufficient to uncover correctly multi-scale communities in a graph. We test the method on a graph benchmark having hierarchical communities, before applying it to a real social network measured in a primary school.

**Index Terms**—Graph wavelets, multiscale community mining

## I. INTRODUCTION

Many data are represented as networks (or weighted graphs), for instance social networks, networks of sensors, of computers (Internet), of neurons, etc... One frequent and striking feature is their modular structure, i.e., there exist groups of nodes, called communities [1], that are more connected with themselves than with the rest of the network. Nodes in a community often sharing properties, community mining provides both a sketch of the structure of a network, and some insight on nodes' properties. Grouping nodes in communities has also been used, for instance, for distributed estimation on networks [2] (for general discussions, see [3]).

One issue in community mining is to decide the scale at which the network is analyzed. Often, works on this topic discard the question of scale or propose only *ad-hoc* discussions. Among several methods (see the review [1]), the optimization of modularity [4] is a popular approach. As it is known to favor an intrinsic scale of description [5], [6], relevant works on this question obtain a scale-dependent community mining by modifying the modularity, either by introducing some *ad-hoc* parameter [7], [8] or using random walk interpretation [9], [10]. Our proposition for community mining is to rely on spectral graph wavelets [11] as a natural way to introduce the notion of scale in graphs for community mining.

The present work elaborates on [12] and develops a scale-dependent procedure which identifies community structures at different scales. The present paper presents two novelties. Firstly, the method will not rely on optimizing a modified modularity as it is done in [12]. Instead, the community detection is tackled as a clustering problem, where wavelets provide scale-dependent features for the nodes and the gap properties of hierarchical clustering are used to decide on

the number of detected clusters (i.e., communities). A second novelty of the present approach is to design a new and faster algorithm using the wavelet transform of a small number of random vectors (taking values on the nodes), for clustering the nodes in communities.

Section II recalls background material on spectral graph wavelets from [11]. The clustering of nodes in communities is described in Section III. Its faster version using the wavelet transform of random vectors is the object of Section IV. The application of this method on a real dataset of social interactions in a primary school is shown in Section V. We conclude in Section VI.

## II. SPECTRAL GRAPH WAVELETS

We review here some material to introduce wavelets on graphs, following [11].

Let  $\mathcal{G} = (V, E, A)$  be a undirected weighted graph with  $V$  the set of  $N$  nodes,  $E$  the set of edges, and  $A$  the weighted adjacency matrix such that  $A_{ij} = A_{ji} \geq 0$  is the weight of the edge between nodes  $i$  and  $j$ . The strengths of nodes are collected in  $D$ , a diagonal matrix with  $D_{ii} = d_i = \sum_{j \neq i} A_{ij}$  the strength of node  $i$ . The normalized Laplacian matrix reads  $\mathcal{L} = D^{-\frac{1}{2}} L D^{-\frac{1}{2}} = I - D^{-\frac{1}{2}} A D^{-\frac{1}{2}}$ .  $\mathcal{L}$  is real symmetric, therefore diagonalisable: its spectrum is composed of  $(\lambda_l)_{l=1 \dots N}$  its set of eigenvalues that we decide to sort in ascending order from  $\lambda_1 = 0$  to  $\lambda_N \leq 2$ ; and of  $\chi$  the matrix of its normalized eigenvectors:  $\chi = (\chi_1 | \chi_2 | \dots | \chi_N)$ , where each  $\chi_l$  is a vector in  $\mathbb{R}^N$ . The normalized Laplacian  $\mathcal{L}$  is preferred here to the non-normalized one  $L = D - A$ , because of its relations with community detection [1].

By analogy to properties of the continuous Laplacian operator,  $\chi$  can be considered as the matrix of the graph's Fourier modes, and  $(\sqrt{\lambda_l})_{l=1 \dots N}$  its set of associated "frequencies" (note that other notions of Fourier transforms on graphs could be used, e.g. [13]). The graph Fourier transform  $\hat{f}$  of a signal  $f$  defined on the nodes of the graph reads:  $\hat{f} = \chi^\top f$ . For instance, a Dirac impulse  $\delta_a$  localized in node  $a$  has a Fourier transform of components  $\hat{\delta}_a(l) = \chi_l(a)$  for  $l = 1, \dots, N$ .

Spectral graph wavelets were defined in [11]. Let us define a band-pass filter kernel  $g(\cdot)$ . For a scale parameter  $s > 0$ , let us use the stretched band-pass filter  $g(s \cdot)$ , whose matrix representation on the graph Fourier modes is  $\hat{G}_s = \text{diag}(g(s\lambda_1), \dots, g(s\lambda_N))$ . To define wavelet  $\psi_{s,a}$  at scale  $s$  and around node  $a$ , one applies this filter on a Dirac impulse  $\delta_a$

in node  $a$ . It turns out that the wavelet basis at scale  $s$  reads in the graph domain as:  $\Psi_s = (\psi_{s,1}|\psi_{s,2}|\dots|\psi_{s,N}) = \chi \hat{G}_s \chi^\top$ . The wavelet transform of a signal  $f$  on the graph at scale  $s$  and node  $a$  is obtained as  $\psi_{s,a}^\top f$  and, in matrix form, its wavelet transform on all nodes is the vector  $\Psi_s f$ .

Note that, at small scales (small scale parameter  $s$ ), the filter  $g(s \cdot)$  lets through high frequency modes essential to good localization: corresponding wavelets extends only to their close neighborhood in the graph. At large scales (large  $s$ ) the filter is compressed around low frequency modes and wavelets encode a wider and coarser local environment.

The details of the band-pass kernel  $g$  follow [11] with the modifications reported in [12]. They are recalled in III-A.

### III. MULTISCALE COMMUNITY MINING

Our proposition is to tackle community detection in a network represented by a graph  $\mathcal{G}$  as a problem of detecting clusters of nodes that have the same neighborhood at a chosen scale of analysis. By construction, the wavelet associated to a node  $a$  and a scale  $s$  is local, centered around this node and it spreads on its neighborhood so that the larger is  $s$ , the larger is the spanned neighborhood. If two nodes are in the same community, their wavelets will overlap as soon as the scale of analysis corresponds to the size of the community. In some sense, wavelets give an ‘‘ego-centered’’ view of how a node ‘‘sees’’ the network at that scale. Taking advantage of this local information encoded in wavelets, we develop an approach that clusters together nodes whose local environments are similar, i.e., whose associated wavelets are correlated. A multiscale description of the communities will be achieved by analysing the network with different scale parameters  $s$ .

The proposed method is detailed in Section III-A, and an example on a graph benchmark is given in Section III-B.

#### A. Basic method for multiscale community mining

The method consists in four steps:

**1. Design of wavelets and choice of scales.** The band-pass filter kernel  $g$  from [11] is used:

$$g(x; \alpha, \beta, x_1, x_2) = \begin{cases} x_1^{-\alpha} x^\alpha & \text{for } x < x_1 \\ p(x) & \text{for } x_1 \leq x \leq x_2 \\ x_2^\beta x^{-\beta} & \text{for } x > x_2. \end{cases} \quad (1)$$

$p(x)$  is taken as the unique cubic polynomial interpolation that respects the continuity of  $g$  and its derivative  $g'$ . Parameters are set according to the discussion in 3.1 of [12]:  $x_1 = 1$  and  $\alpha = 2$ ,  $x_2 = \lambda_{N-1}/\lambda_2$  and  $\beta = 1/\log_{10}\left(\frac{\lambda_3}{\lambda_2}\right)$ . For community mining, this ensures that eigenvector  $\chi_2$  (called the Fiedler vector and known to be important for communities) is seen by wavelets at all scales if one uses  $s_{min} = 1/\lambda_2$  and  $s_{max} = \lambda_{N-1}/\lambda_2^2$  as bounds for the scaling parameter. A sampling of scales in  $[s_{min}, s_{max}]$  is then decided upon. For each scale  $s$ , iterate the next three steps.

**2. Compute wavelets.** At scale  $s$ , the feature vector associated to each node  $a$  is the wavelet  $\psi_{s,a}$  after normalization in energy. Let us write  $\widetilde{\psi}_{s,a}$  the normalized wavelets.

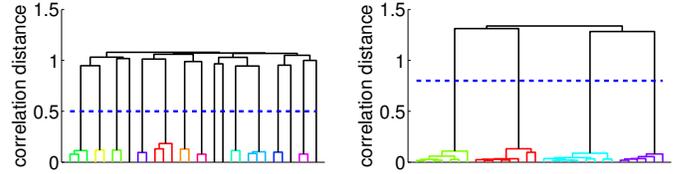


Fig. 1. The method outputs one dendrogram per scale. Each dendrogram is cut at the maximal gap between two of its nodes to obtain the desired partition in communities. Left: dendrogram obtained at a small scale. Right: dendrogram at a large scale. The horizontal dashed line represents the cut.

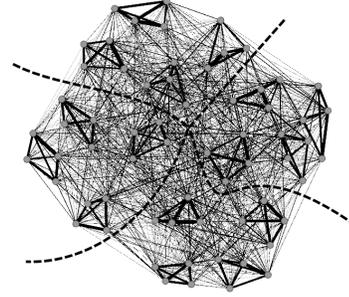


Fig. 2. Sketch of the benchmark graph as defined in section III-B: each node displayed is in fact a community of 10 nodes. The thickness of each link is proportional to the total number of links between the two corresponding communities. The dashed lines are a visual guide to the existence of a large scale structure in four communities.

**3. Compute the correlation distance of wavelets.** Note that, as we use the normalized Laplacian, the relevant mean on the graph of any signal  $f$  defined on the nodes reads:

$$\bar{f} = \chi_1^\top f = \frac{1}{\sqrt{\sum_i d_i}} \sum_{i=1}^N \sqrt{d_i} f(i).$$

With this definition of the mean,  $\forall (s, a) \quad \psi_{s,a}^- = 0$ , and the correlation distance between two feature vectors (i.e., wavelets) associated to nodes  $a$  and  $b$  simply reads:  $D_s(a, b) = 1 - (\widetilde{\psi}_{s,a})^\top \widetilde{\psi}_{s,b}$ . This correlation distance is used to quantify the closeness between nodes  $a$  and  $b$ .

**4. Clustering algorithm.** A hierarchical ‘‘average-linkage’’ clustering algorithm [14], [15] is run on the correlation distance matrix  $D_s(a, b)$ . Its output is a dendrogram whose successive subdivisions code different possible clusterings. As we do not know beforehand how many communities there are in the network at scale  $s$ , we use a method inspired by the gap statistics [16]: the dendrogram is cut at the maximal gap between two of its nodes and we keep the resulting partition as the community structure for scale  $s$ . An illustration of dendrograms and their cut at maximal gap is given in Fig. 1.

Repeating steps 2 to 4 for all scales under study will output the proposed communities at the different scales.

#### B. Example on a controlled benchmark

Following [10], the model of graph defined in [17], for which the multiscale community structure is known, is adopted as a benchmark to test multiscale community mining. We will refer to these graphs as Sales-Pardo (SP) graphs. The

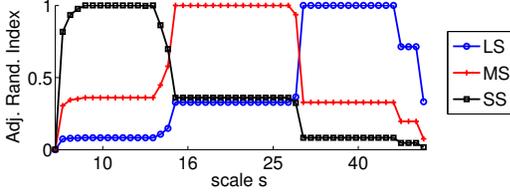


Fig. 3. Adjusted Rand (AR) index of similarity between the partitions found by the algorithm at each scale and the theoretical small scale (black squares), medium scale (red crosses) and large scale (blue circles) partitions. Calculations were done on the Sales-Pardo graph shown in Fig. 2.

global density of links of a SP graph is controlled by a first parameter  $\bar{k}$  (which is the mean degree of the nodes). Then, the intra-community and inter-community relative density of links is fixed by a parameter  $\rho$ . As an example, we choose  $\rho = 1$  and  $\bar{k} = 16$  and a graph of 640 nodes, divided in three hierarchical levels: there are 64 small communities of 10 nodes each (the finest scale) embedded in 16 communities of 40 nodes each (the intermediate scale), themselves embedded in 4 communities of 160 nodes each (the coarsest scale). A realization of the network is visualized in Fig. 2.

Fig. 3 shows that the method successfully recovers the three levels of description. Indeed, for each scale  $s$ , we plot the Adjusted Rand (AR) index [18], which measures the similarity between the partition found by the algorithm at that scale and the three known theoretical partitions. We observe plateaux at 1 that indicate that the three levels of communities are uncovered within large intervals of scales.

#### IV. USING RANDOM VECTORS FOR COMMUNITY MINING IN NETWORKS

The proposed method is well suited for networks of around one thousand nodes. Beyond this, the computational cost becomes prohibitive because of two steps: the computation of the wavelets and the computation of the correlation distance matrix. For the wavelets, one way of going to larger networks is to bypass the full diagonalisation of the Laplacian matrix; this was proposed by Hammond et. al [11] by using Chebychev polynomials and we do not detail that here. For the correlation, we propose a way to bypass its exact computation by considering the wavelet transform of a few random signals.

##### A. Exact correlation distance

The wavelet at scale  $s$  centered around node  $a$  can be written  $\psi_{s,a} = \chi \hat{G}_s \chi^\top \delta_a$ . Their average being null, the correlation distance between the wavelet centered around node  $a$  and the one centered around node  $b$  (at scale  $s$ ) simply reads:

$$D_s(a, b) = 1 - (\widetilde{\psi_{s,a}})^T \widetilde{\psi_{s,b}} = 1 - \frac{\delta_a^\top \chi \hat{G}_s^2 \chi^\top \delta_b}{\|\psi_{s,a}\|_2 \|\psi_{s,b}\|_2}.$$

##### B. Approximate correlation distance using random vectors

Consider a random vector  $r$  defined on the nodes of the graph, taking at each node i.i.d. values of zero mean and finite variance  $\sigma^2$ . Define the feature vector at scale  $s$  associated to node  $a$  as:  $f_{s,a} = \psi_{s,a}^\top r$  (here at one dimension because

we consider for now only one random vector). Moreover, define  $f_s = (f_{s,1} | f_{s,2} | \dots | f_{s,N}) = r^\top \chi \hat{G}_s \chi^\top$ . Because  $r$  is a random vector centered around 0, its expected value is null, and:  $\mathbb{E}(f_s) = \mathbb{E}(r)^\top \chi \hat{G}_s \chi^\top = 0$ . Therefore, the correlation between the feature vector associated to node  $a$  and the one associated to node  $b$  simply reads:

$$\text{Cor}(f_{s,a}, f_{s,b}) = \frac{\delta_a^\top f_s^\top f_s \delta_b}{\|f_{s,a}\|_2 \|f_{s,b}\|_2} = \frac{\delta_a^\top \chi \hat{G}_s \chi^\top r r^\top \chi \hat{G}_s \chi^\top \delta_b}{\|f_{s,a}\|_2 \|f_{s,b}\|_2}.$$

The expected value of the numerator gives:

$$\begin{aligned} \mathbb{E}(\delta_a^\top \chi \hat{G}_s \chi^\top r r^\top \chi \hat{G}_s \chi^\top \delta_b) &= \delta_a^\top \chi \hat{G}_s \chi^\top \mathbb{E}(r r^\top) \chi \hat{G}_s \chi^\top \delta_b \\ &= \sigma^2 \delta_a^\top \chi \hat{G}_s^2 \chi^\top \delta_b. \end{aligned}$$

Moreover:

$$\begin{aligned} \mathbb{E}(\|f_{s,a}\|_2^2) &= \mathbb{E}(f_{s,a}^\top f_{s,a}) = \delta_a^\top \chi \hat{G}_s \chi^\top \mathbb{E}(r r^\top) \chi \hat{G}_s \chi^\top \delta_a \\ &= \sigma^2 \delta_a^\top \chi \hat{G}_s^2 \chi^\top \delta_a = \sigma^2 \|\psi_{s,a}\|_2^2. \end{aligned}$$

This implies:

$$\mathbb{E}(1 - \text{Cor}(f_{s,a}, f_{s,b})) = 1 - \frac{\sigma^2 \delta_a^\top \chi \hat{G}_s^2 \chi^\top \delta_b}{\sigma \|\psi_{s,a}\|_2 \sigma \|\psi_{s,b}\|_2} = D_s(a, b).$$

Now, this calculation is true when considering the expected value of one random vector. We will approximate this expected value by taking the average over  $n_s$  random vectors.

Consider  $n_s$  i.i.d. random vectors  $r_l$ , of zero mean and finite variance  $\sigma^2$  and  $\mathbf{r} = (r_1 | r_2 | \dots | r_{n_s})$  the matrix whose columns are these vectors. Define the feature vector associated to node  $a$  as  $f_{s,a} = \psi_{s,a}^\top \mathbf{r}$  (its dimension is now  $n_s$ ); and the feature matrix  $\mathbf{f}_s$  whose columns are these feature vectors:

$$\mathbf{f}_s = (f_{s,1} | f_{s,2} | \dots | f_{s,N}) = \mathbf{r}^\top \chi \hat{G}_s \chi^\top. \quad (2)$$

As  $n_s$  tends to infinity, the correlation of the feature vectors tends to the exact correlation matrix. In practice, we show in Section IV-C that a relatively small  $n_s$  compared to  $N$  is sufficient. The feature matrix is thereby smaller (of size  $N \times n_s$  instead of  $N^2$ ) and the computation of the correlations faster.

##### C. Method with random vectors and illustration

For large networks, or for faster computation, one will use the method of III-A with the proposed changes: in Step 2, features are those from Eq. (2); in Step 3, the exact correlation distance  $D_s(a, b)$  is replaced by  $1 - \text{Cor}(f_{s,a}, f_{s,b})$ . The rest of the algorithm is unchanged.

To test the efficiency of the method we propose the following protocole. Generate 100 Sales-Pardo random graphs as described in Section III-B. Define the small (resp. medium, large) scale recall ratio as the maximum Adjusted Rand index between the theoretical small (resp. medium, large) scale partition and all the partitions found by the algorithm. We plot in Fig. 4 (top) the average of the three ratios on the 100 graphs with respect to  $n_s$ , the number of random signals used. We observe that with a very few random signals we are able to uncover the theoretical multiscale structure of the network. As expected, uncovering the small scale partition requires more

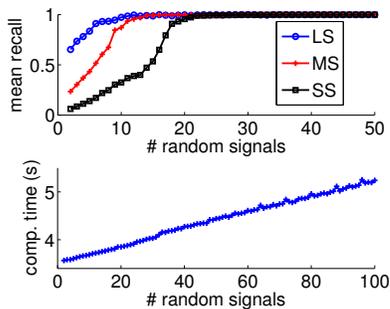


Fig. 4. Top: small, medium and large scale recall ratios with respect to  $n_s$ , the number of random vector used. Bottom: computation time in seconds for the full community mining procedure with respect to  $n_s$ . Results are averaged over 100 realisations of a Sales-Pardo graph, as described in section III-B.

vectors than finding the medium or large scale partitions. In average on this benchmark, 25 random vectors are necessary to uncover all levels of description of the network, instead of the 640 wavelets. Fig. 4 (bottom) shows the average computation time of the algorithm with respect to  $n_s$ : it is shorter than the 16 seconds required by the classical method.

## V. SOCIAL CONTACTS IN A PRIMARY SCHOOL

We propose to apply this method on a real graph of social interactions between 242 children and teachers of a primary school [19]. This graph was measured by the Sociopatterns [20] collaboration using RFID tags. There are two classes in each grade, and grades range from 1<sup>st</sup> to 5<sup>th</sup> grade. Results with  $n_s = 30$  random vectors are shown in Fig. 5. We detect three major partitions at different scales. For  $s \geq 42$ , a large scale partition is uncovered with two communities (grades 1 to 3 on one side, and grades 4 and 5 on the other). For  $27 \leq s \leq 39$ , a medium scale partition is uncovered where nodes are separated with respect to their grades. Finally, for  $14 \leq s \leq 22$ , the algorithm outputs a small scale partition where nodes are separated with respect to their classes. Results for smaller scales are not shown for clarity's sake, but they typically show that classes are separated in groups of friends.

## VI. CONCLUSION

In an effort to speed up the calculations required by the method presented in [12], we propose two improvements. The first one is that we no longer cut the dendrogram by optimizing a scale-dependent modularity function but we simply cut the dendrogram at its maximal gap. More important, the second improvement relies on the wavelet transform of a few random signals to bypass the calculation of the full correlation matrix. Instead of computing the correlation of  $N$  wavelets of size  $N$ , we only need to compute the correlation of  $N$  feature vectors of size  $n_s$ . We show on a graph benchmark and on a real world graph that the number of necessary random vectors for a full recovery can be small compared to  $N$ . The use of random vectors implies that this community mining method is not deterministic anymore, and, in future work, we will take advantage of this to define a notion of stability of a partition found at a given scale.

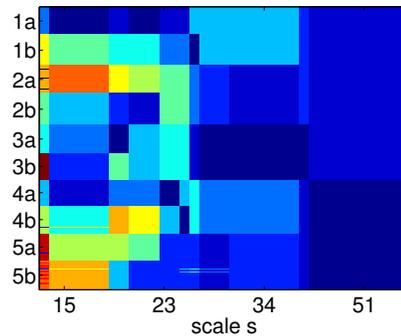


Fig. 5. Multiscale communities of a graph of social interactions measured in a primary school. This matrix has 242 lines (one per participant in the experiment). For each scale (for each column of the matrix), two nodes have the same color if they are in the same community at that scale. Nodes are sorted with respect to their class, which are recalled on the left. For instance, “1a” stands for “1<sup>st</sup> grade class A”.

## REFERENCES

- [1] S. Fortunato, “Community detection in graphs,” *Physics Reports*, vol. 486, no. 3-5, pp. 75–174, 2010.
- [2] F. Cattivelli and A. H. Sayed, “Hierarchical diffusion algorithms for distributed estimation,” *Proc. IEEE Workshop on Statistical Signal Processing (SSP)*, pp. 537–540, 2009.
- [3] “Special section – adaptation and learning over complex networks,” *Signal Processing Magazine*, vol. 30, no. 3, 2013.
- [4] M. Newman, “Modularity and community structure in networks,” *PNAS*, vol. 103, no. 23, p. 8577, 2006.
- [5] S. Fortunato and M. Barthelemy, “Resolution limit in community detection,” *PNAS*, vol. 104, no. 1, p. 36, 2007.
- [6] J. Kumpula, J. Saramäki, K. Kaski, and J. Kertesz, “Limited resolution in complex network community detection with Potts model approach,” *Eur. Phys. J. B*, vol. 56, no. 1, pp. 41–45, 2007.
- [7] J. Reichardt and S. Bornholdt, “Statistical mechanics of community detection,” *Physical Review E*, vol. 74, no. 1, p. 016110, 2006.
- [8] A. Arenas, A. Fernandez, and S. Gomez, “Analysis of the structure of complex networks at different resolution levels,” *New Journal of Physics*, vol. 10, no. 5, p. 053039, 2008.
- [9] M. Schaub, J. Delvenne, S. Yaliraki, and M. Barahona, “Markov dynamics as a zooming lens for multiscale community detection: non clique-like communities and the field-of-view limit,” *PLoS one*, vol. 7, no. 2, p. e32210, 2012.
- [10] R. Lambiotte, “Multi-scale modularity in complex networks,” in *Proc. 8th Int. Symp. WiOpt*, 2010, pp. 546–553.
- [11] D. Hammond, P. Vandergheynst, and R. Gribonval, “Wavelets on graphs via spectral graph theory,” *Applied and Computational Harmonic Analysis*, vol. 30, no. 2, pp. 129–150, 2011.
- [12] N. Tremblay and P. Borgnat, “Multiscale community mining in networks using spectral graph wavelets,” *EUSPICO*, 2013.
- [13] A. Sandryhaila and J. Moura, “Discrete signal processing on graphs,” *IEEE Transactions on Signal Processing*, vol. 61, no. 7, 2013.
- [14] A. Jain, M. Murty, and P. Flynn, “Data clustering: a review,” *ACM computing surveys (CSUR)*, vol. 31, no. 3, pp. 264–323, 1999.
- [15] B. King, “Step-wise clustering procedures,” *Journal of the American Statistical Association*, vol. 62, no. 317, pp. 86–101, 1967.
- [16] T. Hastie, R. Tibshirani, J. Friedman *et al.*, “The elements of statistical learning: data mining, inference, and prediction,” 2001.
- [17] M. Sales-Pardo, R. Guimera, A. Moreira, and L. Amaral, “Extracting the hierarchical organization of complex systems,” *PNAS*, vol. 104, no. 39, pp. 15 224–15 229, 2007.
- [18] L. Hubert and P. Arabie, “Comparing partitions,” *Journal of classification*, vol. 2, no. 1, pp. 193–218, 1985.
- [19] J. Stehle, N. Voirin, A. Barrat, C. Cattuto *et al.*, “High-resolution measurements of face-to-face contact patterns in a primary school,” *PLoS one*, vol. 6, no. 8, p. e23176, 2011.
- [20] www.sociopatterns.org.