Graph Wavelets for Multiscale Community Mining

Nicolas Tremblay, Pierre Borgnat *Member, IEEE,*
Laboratoire de Physique, CNRS, ENS de Lyon, Université de Lyon, France

***Abstract*—We develop a signal processing approach to the multiscale detection of communities in networks, that is of groups of nodes well connected together. The method relies on carefully engineered wavelets on graphs to introduce the notion of scale and to obtain a local view of the graph from each node. Computing the correlations between wavelets centered at different nodes, one has access to a notion of similarity between nodes, thereby enabling a clustering procedure that groups nodes according to their community at the scale of analysis. By using a collection of random vectors to estimate the correlation between the nodes, we show that the method is suitable for the analysis of large graphs. Furthermore, we introduce a notion of partition stability and a statistical test allowing us to assess which scales of analysis of the network are relevant. The effectiveness of the method is discussed first on multiscale graph benchmarks, then on real data of social networks and on models for signal processing on graphs.**

***Index Terms*—Graph wavelets, community mining, multiscale community, spectral graph theory, wavelet transform**

I. Introduction

In many complex systems, data are naturally represented as networks (or weighted graphs): social networks, sensor networks, Internet networks, neuronal networks, transportation networks, biological networks. . . [5] A striking property of many networks, and a common way of simplifying the network's analysis, is their modular structure, i.e. there exists groups of nodes, called communities [6], that are more connected with themselves than with the rest of the network. As nodes in a same community tend to share common properties, a partition of the network in communities may provide both a sketch of the structure of the network, and some insight into the properties of the nodes, for signal processing issues especially [7]. In network science, state of the art of community detection (see the review [6]) is often based on the optimisation over the possible partitions of nodes of a suitable criterion, such as the popular modularity [8] or other criteria such as the normalized cut [9]; or by mapping random walks from an information theory point of view such as in Ref. [10].

Often, the structural reality of a network is a superposition of several partitions in communities at different scales, with, for instance, small communities with only a few nodes, embedded in larger communities. Examples of these are shown later on in Fig. 3 for a classical benchmark of complex networks, Fig. 9 for a real-world network of social interactions, and Fig. 11 for a toy-model sensor network. One could add other examples of networks displaying this property, such as connectivity networks in the brain [11], [12] or metabolic networks [13]. The issue of the scale of description is usually implicitly discarded as soon as an algorithm is asked to output only one partition as a representation of an often complex structural reality. In fact, the scale is generally not chosen by the user but arbitrarily imposed by the algorithm. For instance, this has been shown for algorithms based on modularity optimisation which favor an intrinsic scale of description [14], [15]. To deal with this issue and to propose a more comprehensive description of a graph's community structure, some authors have proposed multiscale community mining algorithms that output one partition per scale of description. They are either based on random walk processes [16], [17], on definitions of parametric modularities [18], [19], or simply by studying the different solutions of an agglomerative clustering algorithm [9]. These methods have various notions of scale parameters: the strength of the added self-loops in [19], the Markov time in [16], or the loop-number of an agglomerative algorithm in [9]. It is our goal in this paper to design a new multiscale method, deeply rooted in signal processing. In fact, community detection being a central tool of complex graph analysis that informs about the structure of many networks, we believe it is of importance to tackle this problem with the methods of the emerging field of graph signal processing [5]. Communities in a graph may be considered as inhomogeneities, and we will explore throughout this paper how one may use graph wavelets to detect them.

A first contribution of the present work takes advantage of graph wavelets to obtain a scale-dependent analysis of communities in networks. There are several frameworks to introduce wavelets on graphs [20], [21], [22]. We use the one of Hammond et al. [20] based in spectral graph theory of the Laplacian, for three reasons: the relevance of the Laplacian in community detection [6], [7], the easiness with which we can force the wavelet to be sensitive to communities, and the existence of a fast wavelet transform that will be used for fast community mining and detection of stable communities. By construction, a wavelet associated to a node is local in the graph: it is centered around this node and spreads on its neighbourhood so that the larger the scale is, the larger is the spanned neighbourhood. Wavelets on graphs provide an "egocentered" 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 second contribution of this work is a way to apply the method to larger networks (e.g. ten thousand nodes) by

This work was partially supported by the GDR ISIS and the CNRS/JSPS Grants "Benchmarking and statistical analysis tools for modern Internet and sensor network traffics," 2010 and 2012. Preliminary approaches of this work were presented at GlobalSIP 2013 [1], GRETSI 2013 [2], ECCS 2013 [3], and EUSIPCO 2013 [4].

computing the wavelet transform of a few random vectors to estimate the wavelets' correlation.

This work's third contribution is a novel way to assess the stability of the communities, hence their relevance. Indeed, multiscale community mining is the sum of two challenges: output a partition per scale, and assess each of these partitions' relevance. Defining a stability measure of a partition in communities, and a statistical test comparing the studied graph to randomised ones, we develop a way to detect at which scales the network has a relevant community structure.

A preliminary version of this work is presented in [4] where a filtered modularity is used to find the best communities. Here, the optimisation of a filtered modularity function is no longer used and we propose a new, simpler and faster method to cut the dendrogram given by the hierarchical clustering algorithm, inspired by the gap statistics method [23] and detailed in Section IV.

The paper is organized as follows. Section II recalls background material on spectral graph theory and graph wavelets. Some developments of the use of spectral graph wavelets for community mining are presented in Section III: we discuss that once a network is given, a proper choice of scale boundaries ends up with parameters for the band-pass filter defining the wavelets that are different from [20]. Section IV describes the multiscale community mining procedure. Section V shows how one may use the wavelet transform of random vectors to speed up the algorithm and enable the analysis of larger networks. Section VI describes how to detect stable communities by measuring the stability of all uncovered partitions. Section VII discusses a statistical test that enables an automatic detection of scales for which the associated partitions in communities are relevant. Performance obtained with the proposed method is illustrated and discussed on two benchmark networks from the literature in Section VIII and compared to other methods. Then, in Section IX, we successfully apply this method to several examples. We conclude in Section X.

Notations: The following notations will be used. Vectors are denoted by boldface lowercase letters such as a graph signal $\boldsymbol{f}$. Matrices are denoted by boldface capital letters such as the adjacency matrix $\boldsymbol{A}$. Ensembles are denoted by capital letters in calligraphic style such as the ensemble of nodes $\mathcal{V}$. Scalars are denoted either by lowercase letters such as the eigenvalues λ_i, or by capital letters such as the number of nodes N.

II. Spectral Graph Theory and Wavelets

A. The Graph Fourier Transform

Let $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{A})$ be a undirected weighted graph with $\mathcal{V}$ the set of nodes, $\mathcal{E}$ the set of edges, and $\mathbf{A}$ the weighted adjacency matrix such that $\mathbf{A}_{ij} = \mathbf{A}_{ji} \geq 0$ is the weight of the edge between nodes i and j. Note N the total number of nodes. Let us define the graph's Laplacian matrix $\mathbf{L} = \mathbf{D} - \mathbf{A}$ where $\mathbf{D}$ is a diagonal matrix with $\mathbf{D}_{ii} = \mathbf{d}_i = \sum_{j \neq i} \mathbf{A}_{ij}$ the strength of node i. The normalized Laplacian matrix reads $\boldsymbol{\mathcal{L}} = \mathbf{D}^{-\frac{1}{2}}\mathbf{L}\mathbf{D}^{-\frac{1}{2}} = \mathbf{I}_N - \mathbf{D}^{-\frac{1}{2}}\mathbf{A}\mathbf{D}^{-\frac{1}{2}}$, where $\mathbf{I_N}$ is the identity matrix of size N. $\boldsymbol{\mathcal{L}}$ is real symmetric, therefore diagonalizable: its spectrum is composed of $(\lambda_l)_{l=1\ldots N}$ its set of eigenvalues that we sort: $0 = \lambda_1 \leq \lambda_2 \leq \lambda_3 \leq \cdots \leq \lambda_N \leq 2$ [24]; and of $\boldsymbol{\chi}$ the matrix of its normalized eigenvectors: $\boldsymbol{\chi} = (\boldsymbol{\chi}_1|\boldsymbol{\chi}_2|\ldots|\boldsymbol{\chi}_N)$. Considering only connected graphs, the multiplicity of eigenvalue $\lambda_1 = 0$ is 1 [24]. By analogy to the continuous Laplacian operator whose eigenfunctions are the continuous Fourier modes and eigenvalues their squared frequencies, $\boldsymbol{\chi}$ is considered as the matrix of the graph's Fourier modes, and $\left(\sqrt{\lambda_l}\right)_{l=1\ldots N}$ its set of associated "frequencies". A more comprehensive discussion is in Ref. [25]. For instance, the graph Fourier transform $\hat{\boldsymbol{f}}$ of a signal $\boldsymbol{f}$ defined on the nodes of the graph reads: $\hat{\boldsymbol{f}} = \boldsymbol{\chi}^\top \boldsymbol{f}$. Note that other definitions of Fourier vectors could be considered, for example [26]. However, using Fourier vectors defined with the Laplacian is relevant for community detection as its has been widely used in spectral clustering [6]. There is an on-going debate over which version of the Laplacian (normalized or not) should be used. Donetti et al. [27] show their spectral algorithm is more efficient with the normalized Laplacian, without proposing an explanation. More generally, it has been proved that the spectrum of the normalized Laplacian has very close links with famous graph invariants such as the Cheeger constant or the discrepancy, or that λ_2, for instance, is related with the speed of convergence of random walks on the graph [24]. Moreover, the fact that the spectrum is bounded between 0 and 2 generally makes calculations involving the normalized Laplacian easier. For all these reasons, we choose here to use the normalized Laplacian.

B. Spectral Graph Wavelets

Spectral graph wavelets were defined in [20] using the graph Fourier modes previously defined. In the following, we write the theory of [20] in the more condensed language of linear algebra. Let us note $\boldsymbol{\psi}_{s,a}$ the wavelet at scale $s \in \mathbb{R}_+^*$ centered around node $a \in \mathcal{V}$. Its construction is based on band-pass filters defined in the graph Fourier domain, generated by stretching a unique band-pass wavelet filter kernel $g(\cdot)$ by a scale parameter $s > 0$. The stretched filter has a matrix representation $\boldsymbol{G}_s = \operatorname{diag}(g(s\lambda_1), \ldots, g(s\lambda_N))$ that is diagonal on the Fourier modes (the N eigenvectors of $\boldsymbol{\mathcal{L}}$). Hence, the wavelet basis at scale s reads as:

$$\boldsymbol{\Psi}_s = (\boldsymbol{\psi}_{s,1}|\boldsymbol{\psi}_{s,2}|\ldots|\boldsymbol{\psi}_{s,N}) = \boldsymbol{\chi}\boldsymbol{G}_s\boldsymbol{\chi}^\top. \tag{1}$$

For just one wavelet, this reads equivalently as:

$$\boldsymbol{\psi}_{s,a} = \boldsymbol{\chi}\boldsymbol{G}_s\boldsymbol{\chi}^\top\boldsymbol{\delta}_a. \tag{2}$$

Then the wavelet coefficient at scale s and node a of a signal $\boldsymbol{f}$ reads $W_f(s,a) = \boldsymbol{\psi}_{s,a}^\top \boldsymbol{f}$. Our first use will be of the localized wavelets $\boldsymbol{\psi}_{s,a}$ themselves, and the wavelet transform of signals is required later on in Section V. Note also that the filter kernel function g is defined as a *continuous* function defined on $\mathbb{R}^+$ and sampled on the graph Fourier space. On the other hand, for each given scale parameter s, the filter $\boldsymbol{G}_s$ is *discrete*: only the values of $g(s\cdot)$ on the spectrum $(\lambda_l)_{l=1\ldots N}$ are needed, hence the matrix notation. However, the wavelets are continuous in scale s and discrete in space: $\boldsymbol{\psi}_{s,a}$ is a column vector of size N giving its value at each node.

The intuition behind this definition of wavelets on graphs is that, at small scales (small s), the filter $g(s\,\cdot)$ is stretched out.

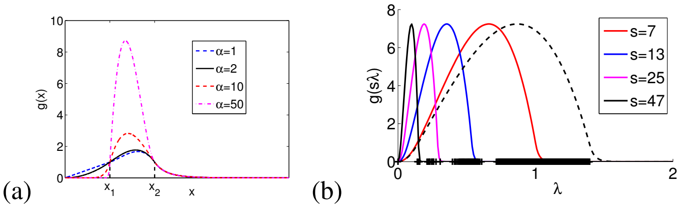


Fig. 1. (a) Shape of the filter function $g(x)$ for four different values of α. (b) Band-pass wavelet filter functions g for $M = 4$ different scales and $x_1 = 1$, $x_2 = 7$, $\alpha = 2$ and $\beta = 41$. The actual eigenvalues λ_i of the network (the multi scale SP benchmark used in section VIII) with $N = 640$ nodes are indicated with crosses on the x axis. A fifth filter function is shown with a dashed line: it corresponds to a scale even smaller than s_{min}.

It thus lets through high frequency modes essential to good localization; the corresponding wavelets extends only to their close neighbourhood in the graph. At large scales (large s) the filter function is compressed around low frequency modes and this creates wavelets encoding a coarser description of the local environment.

The parameters defining the precise shape of the band-pass wavelet filter kernel $g(\cdot)$ are important. Part of our contribution is to engineer a suitable filter kernel for community detection: details are given in Section III.

C. Fast Wavelet Transform

Beyond a size of 1000 nodes, the computational cost of the Laplacian's diagonalization becomes prohibitive: the exact computation of the Fourier matrix $\boldsymbol{\chi}$ is no longer possible. Hammond et. al [20] designed an efficient way of bypassing the diagonalisation of the Laplacian and obtained an approximation of the wavelets by using Chebychev polynomials to approximate the filters [28]. We will write $\mathcal{FWT}_s$ the operator corresponding to this fast wavelet transform at scale s. For a given signal $\boldsymbol{f}$, $\mathcal{FWT}_s(\boldsymbol{f})$ is a vector where the element $\mathcal{FWT}_s(\boldsymbol{f})(i)$ is the wavelet coefficient of $\boldsymbol{f}$ at node i and scale s. Then the wavelet basis $\boldsymbol{\Psi}_s$ at scale s can be efficiently approximated by:

$$\boldsymbol{\Psi}_s \sim \mathcal{FWT}_s(\boldsymbol{I}_N), \tag{3}$$

where $\boldsymbol{I}_N$ is the identity matrix of size N. The error of approximation is tuned by the degree m of the Chebychev polynomial: the larger is m, the better is the approximation. Unless otherwise specified, we use in the following $m = 50$.

III. Graph wavelet filter parameters

We use the band-pass filter kernel g proposed in [20]:

$$g(x; \alpha, \beta, x_1, x_2) = \begin{cases} x_1^{-\alpha} x^{\alpha} & \text{for} \quad x < x_1 \\ p(x) & \text{for} \quad x_1 \leq x \leq x_2 \\ x_2^{\beta} x^{-\beta} & \text{for} \quad x > x_2. \end{cases} \tag{4}$$

where $p(x)$ is taken as the unique cubic polynomial interpolation that respects the continuity of g and its derivative g'. The integers α and β, and the transition points x_1 and x_2 are the parameters of the filter, which we here define in a novel way adapted for community mining. For that, let us study which scale boundaries are relevant. The parameters are based on an argument of spectral clustering of graphs [6], [7]: the eigenvector $\boldsymbol{\chi}_2$ (associated to the smallest non-zero eigenvalue λ_2, also called Fiedler vector) is the first in importance for community mining because it contains information on the coarsest description of the graph. The following describes one proposition for the construction of the band-pass wavelet filter from this argument.

A first consequence is that the maximum scale s_{max} is set so that the filter function $g(s_{max}\, x)$ starts decaying as a power law only after $x = \lambda_2$, hence $s_{max} = x_2/\lambda_2$. We require also that the filter at the maximum scale is highly selective around λ_2; for that, all other eigenmodes (especially λ_3) have to be attenuated. Choosing an attenuation by a factor 10, this leads us to: $g(s_{max}\,\lambda_2) = 10\, g(s_{max}\,\lambda_3)$, hence $\beta = 1/\log_{10}\left(\frac{\lambda_3}{\lambda_2}\right)$. We thereby ensure that the filter at the maximum scale essentially keeps the information from $\boldsymbol{\chi}_2$.

Second, we need to keep a part of $\boldsymbol{\chi}_2$ in the wavelets of every scale, so that all wavelets are sensitive to large scale community structure. We propose as minimum scale s_{min} the one for which $g(s_{min}\,\lambda_2)$ becomes smaller than 1. Using eq. (4), this gives $s_{min} = x_1/\lambda_2$. Imposing also that $g(s_{min}\,\cdot)$ spans at least the whole range of eigenvalues between 0 and 1 (indeed, spectral clustering algorithms always consider only the first few eigenvectors [29], and experimentally we never need to stretch the band-pass further than $\lambda = 1$) we need $s_{min} \times 1 = x_2$.

This argumentation gives us a value for β and three equations linking x_1, x_2, s_{min} and s_{max} :

$$s_{min} = \frac{x_1}{\lambda_2}, \quad x_2 = \frac{x_1}{\lambda_2}, \quad s_{max} = \frac{x_1}{\lambda_2^2}, \tag{5}$$

where we see that x_1 has the unique effect of translating the scale boundaries s_{min} and s_{max} on the $\mathbb{R}^+$ axis. x_1 can therefore be safely fixed to 1. This leaves α, describing the cut-off at low frequency. In classical wavelets, α corresponds to the number of moments equal to zero. But in our case, this interpretation is not valid because the smallest value of $s\lambda$ is $s_{min}\lambda_2 = 1$, and is therefore already too large to be affected by α. In fact, α only has an indirect effect on the maximum of $g(x)$: the larger is α, the larger is the maximum value of $g(x)$, the more selective is the filter between x_1 and x_2 as shown on Fig. 1a. This selectivity is wished for at large scale but this is already insured by the other parameters we fixed. The effect of α will especially be seen at medium and small scales for which we want to keep the information of small eigenvalues: we do not want the filter to be too selective. Moreover, for localisation purposes (see [20]), α needs to be larger than 1. We therefore fix it to 2 in the following. Fig. 1b shows wavelet filters $g(s\cdot)$ with the proposed range of scale and parameters for an example of the Sales-Pardo benchmark network [30] (see Section VIII A).

Finally, we have to choose a sampling of M scales between the scale boundaries: $\mathcal{S} = \{s_1 = s_{min}, s_2, \ldots, s_M = s_{max}\}$. We choose them logarithmically spaced because the density of eigenvalues on the interval $[0, 2]$ is not uniform: they are much more grouped around 1 than 0 for complex graphs with communities [31]. This non-uniformity is indeed observable in Fig. 1b, where the eigenvalues are plotted on the x-axis.

Therefore, a small difference at small scale (a small scale takes into account the largest eigenvalues) has a much bigger impact on the clustering than the same small difference at large scale. As for the number of scales M we decide to scan, we choose, by analogy to the classical 1-D discrete wavelets case: $M = \kappa \log_2(N)$ where κ is typically inferior to 10, and N the number of nodes [25].

IV. Multiscale Community Mining

At a given scale $s \in \mathcal{S}$, the proposed community mining protocol is described in the following three key points. It consists in applying unsupervised classification to a set of scale dependent feature vectors defined by the wavelet transform.

1. Scale-dependent feature vectors. The aim is to group together nodes whose topological environments are similar. As the local information and topology in a graph is encoded in the wavelets we define for each node a its feature vector to be its associated wavelet $\boldsymbol{\psi}_{s,a}$.

2. Correlation distance. To compare nodes, we use a distance between their features, chosen as the correlation distance between the wavelet centered around node a and the one centered around node b (at scale s):

$$\boldsymbol{D}_s(a,b) = 1 - \frac{\boldsymbol{\psi}_{s,a}^\top \boldsymbol{\psi}_{s,b}}{||\boldsymbol{\psi}_{s,a}||_2 ||\boldsymbol{\psi}_{s,b}||_2}. \tag{6}$$

Experimentally, this correlation distance yields better results than, e.g., the Euclidean distance.

Note that this is a correlation distance because the wavelet have zero mean. Indeed, as we use the normalized Laplacian $\boldsymbol{\mathcal{L}}$, the relevant mean of any signal $\boldsymbol{f}$ reads:

$$\bar{\boldsymbol{f}} = \boldsymbol{\chi}_1^\top \boldsymbol{f} = \frac{1}{\sqrt{\sum_i \boldsymbol{d}_i}} \sum_{i=1}^{N} \sqrt{\boldsymbol{d}_i} \boldsymbol{f}(i). \tag{7}$$

With this definition of the mean (used for instance in [20] Section 5.1), we have that: $\forall (s,a) \quad \bar{\boldsymbol{\psi}}_{s,a} = \boldsymbol{\chi}_1^\top \boldsymbol{\psi}_{s,a} = \boldsymbol{\chi}_1^\top \boldsymbol{\chi} \boldsymbol{G}_s \boldsymbol{\chi}^\top \boldsymbol{\delta}_a = g_s(1)\boldsymbol{\chi}_1(a)$ since $\boldsymbol{\chi}$ is an orthonormal matrix. By definition of a wavelet filter, the constant component $g_s(1)$ is null, hence $\bar{\boldsymbol{\psi}}_{s,a} = 0$. Note that if we had used the combinatorial Laplacian $\boldsymbol{L}$, whose first eigenvector is constant, the mean of any signal $\boldsymbol{f}$ defined on the nodes would have been the classical mean: $\bar{\boldsymbol{f}} = \frac{1}{N} \sum_{i=1}^{N} \boldsymbol{f}(i)$.

3. Clustering algorithm. We use a hierarchical "average-linkage" clustering algorithm [32], [33] on this distance matrix $\boldsymbol{D}_s$. This hierarchical algorithm gives a dendrogram as its output that one needs to cut horizontally to obtain the partition P_s (see Fig 2b for an example of a dendrogram of a toy graph with $N = 32$ nodes illustrated in Fig 2a). A main question is: where should we cut this dendrogram? As we do not know beforehand how many clusters there are in the network, we have to define a criterion to cut the dendrogram. In previous works [4], [1], inspired by the gap statistics method [34], we simply used to cut the dendrogram at its maximal gap. Here, we propose a criterion to cut the dendrogram based on averaging the maximal gaps of all the root-leaf paths of the dendrogram, as this method is more robust to outliers.

More precisely, consider a node a and define its dendrogram-path: it is the path between the leaf of the dendrogram corresponding to node a and the root of the dendrogram (the node of the dendrogram that has the highest correlation distance). Fig. 2b shows an example of a dendrogram with, in red, an example of a dendrogram-path. For this node a, one can plot its gap function Γ_a built the following way: follow the dendrogram-path starting at zero correlation distance. For each correlation distance, the path is between two dendrogram nodes: plot the gap between them. The gap function corresponding to the dendrogram-path shown in Fig. 2b is plotted in Fig. 2c. By averaging all gap functions corresponding to all nodes, one obtains the global gap function:

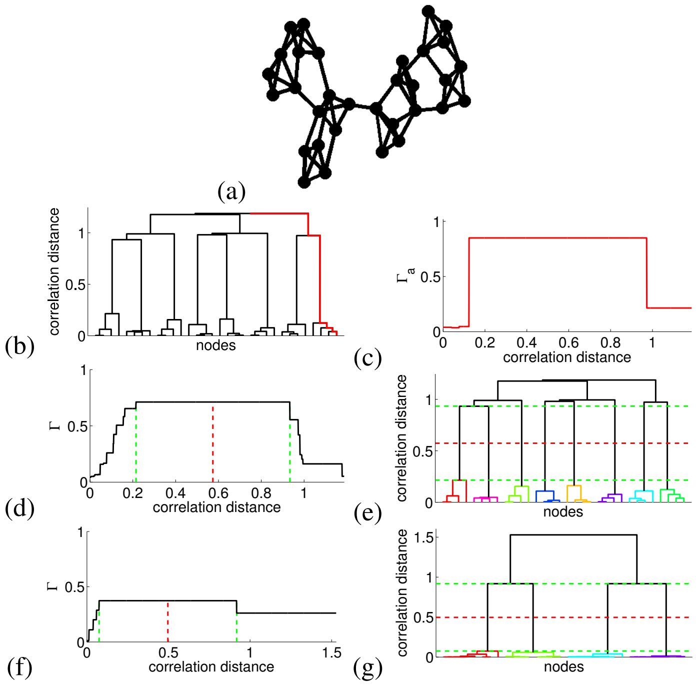


Fig. 2. A toygraph with $N = 32$ nodes is illustrated in (a). (b) shows the dendrogram obtained at an intermediate scale for this graph; in red is an example of dendrogram path corresponding to a node a; (c) shows the associated gap function Γ_a; (d) shows Γ, the average of all gap functions: it is maximal for an interval of correlation distance bounded by the two vertical green dashed lines. The vertical red line is the middle of the interval. (e) shows the effect of the corresponding cut: it separates nodes in communities. (f) and (g) are equivalent to (d) and (e) for a larger scale parameter.

$$\Gamma = \frac{1}{N \text{max(corr. dist.)}} \sum_{a \in \mathcal{V}} \Gamma_a$$

shown in Fig. 2d. Following the gap statistics intuition [34], we consider that the best possible partition given this dendrogram is obtained by cutting the dendrogram at the maximum of Γ (see Fig. 2d and 2e for an illustration). An example at a larger scale is illustrated in Figs. 2f and 2g.

Repeating these three steps for each scale $s \in \mathcal{S}$, one obtains the multiscale set of partitions $\mathcal{P} = \{P_s\}_{s \in \mathcal{S}}$.

V. Fast Community Mining with Random Vectors

At step **1** of the method of Section IV, one needs to compute $\boldsymbol{\Psi}_s$, all wavelets at a given scale s. For that, the fast wavelet transform of N Diracs is sufficient (see Eq. (3)). However, the information that is really needed from these wavelets is their correlation matrix $\boldsymbol{D}_s$ (computed in step **2**). It is possible to bypass the full computation of each wavelet: we propose instead a method to estimate directly the correlation matrix by computing the fast wavelet transform of a few random signals.

Consider a random vector $\boldsymbol{r} \in \mathbb{R}^N$ defined on the nodes of the graph, composed of N independent normal random variables of zero mean and finite variance σ^2. Define the feature $f_{s,a} \in \mathbb{R}$ at scale s associated to node a as the projection of that vector on the wavelet $\boldsymbol{\psi}_{s,a}$:

$$f_{s,a} = \boldsymbol{\psi}_{s,a}^\top \boldsymbol{r} = \sum_{k=1}^{N} \psi_{s,a}(k) r(k). \tag{8}$$

Consider now the correlation of the features associated to nodes a and b. By definition :

$$\text{Cor}(f_{s,a}, f_{s,b}) = \frac{\mathbb{E}((f_{s,a} - \mathbb{E}(f_{s,a}))(f_{s,b} - \mathbb{E}(f_{s,b})))}{\sqrt{\text{Var}(f_{s,a})\text{Var}(f_{s,b})}}. \tag{9}$$

As a sum of N independent normal random variables, $f_{s,a}$ is a normal random variable of expected value

$$\mathbb{E}(f_{s,a}) = \boldsymbol{\psi}_{s,a}^\top \mathbb{E}(\boldsymbol{r}) = 0 \tag{10}$$

and variance

$$\begin{aligned} \text{Var}(f_{s,a}) &= \mathbb{E}((f_{s,a} - \mathbb{E}(f_{s,a}))^2) = \mathbb{E}(f_{s,a}^2) \\ &= \boldsymbol{\psi}_{s,a}^\top \mathbb{E}(\boldsymbol{r}^\top \boldsymbol{r}) \boldsymbol{\psi}_{s,a} = \sigma^2 ||\boldsymbol{\psi}_{s,a}||_2^2. \end{aligned} \tag{11}$$

Therefore, Eq.(9) is rewritten as:

$$\text{Cor}(f_{s,a}, f_{s,b}) = \frac{\mathbb{E}(f_{s,a} f_{s,b})}{\sigma^2 ||\boldsymbol{\psi}_{s,a}||_2 ||\boldsymbol{\psi}_{s,b}||_2}. \tag{12}$$

Compute the covariance :

$$\begin{aligned} \mathbb{E}(f_{s,a} f_{s,b}) &= \mathbb{E}((\boldsymbol{\psi}_{s,a}^\top \boldsymbol{r})(\boldsymbol{\psi}_{s,b}^\top \boldsymbol{r})) \\ &= \mathbb{E}((\sum_{k=1}^{N} \psi_{s,a}(k) r(k))(\sum_{k'=1}^{N} \psi_{s,b}(k') r(k'))) \\ &= \sum_{k \neq k'} \psi_{s,a}(k) \psi_{s,b}(k') \mathbb{E}(r(k) r(k')) \\ &\quad + \sum_{k=1}^{N} \psi_{s,a}(k) \psi_{s,b}(k) \mathbb{E}(r(k)^2) \\ &= \sigma^2 \psi_{s,a}^\top \psi_{s,b}. \end{aligned} \tag{13}$$

Therefore:

$$\text{Cor}(f_{s,a}, f_{s,b}) = \frac{\boldsymbol{\psi}_{s,a}^\top \boldsymbol{\psi}_{s,b}}{||\boldsymbol{\psi}_{s,a}||_2 ||\boldsymbol{\psi}_{s,b}||_2}. \tag{14}$$

The features' correlation is exactly the correlation between the wavelets centered around nodes a and b. Before discussing the estimation of this correlation, let us show that $f_{s,a}$ and $f_{s,b}$ are jointly Gaussian, i.e. that any linear combination $cf_{s,a} + df_{s,b}$ ($(c, d) \in \mathbb{R}^2$) is Gaussian. In fact:

$$cf_{s,a} + df_{s,b} = \sum_{k=1}^{N} (c\psi_{s,a}(k) + d\psi_{s,b}(k)) r(k)$$

is a sum of independent normal random variables, therefore normal.

To estimate the correlation of Eq. (14), we use the classical sample correlation estimator. Consider now η realizations of $\boldsymbol{r}$ and store them in the matrix $\boldsymbol{R} = (\boldsymbol{r}_1 | \boldsymbol{r}_2 | \ldots | \boldsymbol{r}_\eta) \in \mathbb{R}^{N \times \eta}$ where the i-th column $\boldsymbol{r_i}$ is the i-th realization of $\boldsymbol{r}$. Note $f_{s,a}^i = \boldsymbol{\psi}_{s,a}^\top \boldsymbol{r_i}$ the i-th realization of $f_{s,a}$, and concatenate all its η realizations in the feature vector $\boldsymbol{f}_{s,a}$:

$$\boldsymbol{f}_{s,a}^\top = \boldsymbol{\psi}_{s,a}^\top \boldsymbol{R} \qquad (\boldsymbol{f}_{s,a} \in \mathbb{R}^\eta). \tag{15}$$

The sample correlation coefficient estimator between $\boldsymbol{f}_{s,a}$ and $\boldsymbol{f}_{s,b}$ reads:

$$\hat{C}_{ab,\eta} = \frac{(\boldsymbol{f}_{s,a} - \bar{\boldsymbol{f}}_{s,a})^\top (\boldsymbol{f}_{s,b} - \bar{\boldsymbol{f}}_{s,b})}{||\boldsymbol{f}_{s,a} - \bar{\boldsymbol{f}}_{s,a}||_2 ||\boldsymbol{f}_{s,b} - \bar{\boldsymbol{f}}_{s,b}||_2}, \tag{16}$$

where $\bar{\boldsymbol{f}}_{s,a}$ is the constant vector equal to the average of $\boldsymbol{f}_{s,a}$: if $\mathbb{1}$ is the constant vector equal to 1, $\bar{\boldsymbol{f}}_{s,a} = \frac{1}{\eta} \mathbb{1}^\top \boldsymbol{f}_{s,a} \mathbb{1}$.

As $f_{s,a}$ and $f_{s,b}$ are jointly Gaussian, this estimator is asymptotically consistent, hence:

$$\lim_{\eta \to +\infty} \hat{C}_{ab,\eta} = \text{Cor}(f_{s,a}, f_{s,b}) = \frac{\boldsymbol{\psi}_{s,a}^\top \boldsymbol{\psi}_{s,b}}{||\boldsymbol{\psi}_{s,a}||_2 ||\boldsymbol{\psi}_{s,b}||_2}. \tag{17}$$

Therefore:

$$\lim_{\eta \to +\infty} 1 - \hat{C}_{ab,\eta} = \boldsymbol{D}_s(a, b). \tag{18}$$

In practice, experiments will show (see Section VIII-A) that a relatively small η compared to N is sufficient. Therefore, instead of computing the fast wavelet transform of N Diracs to obtain all wavelets and then compute the corresponding correlation matrix of N vectors of size N, one only needs to compute the fast wavelet transform of a small number η of random vectors and then compute the correlation matrix of N vectors of size η.

Let us recap this fast community mining procedure, at a given scale s, in three steps:

1. Generate a matrix of η Gaussian random vectors of zero mean and variance σ^2 (in practice: $\sigma^2 = 1$): $\boldsymbol{R} = (\boldsymbol{r}_1 | \boldsymbol{r}_2 | \cdots | \boldsymbol{r}_\eta) \in \mathbb{R}^{N \times \eta}$. Compute the fast wavelet transform of each of those η vectors to obtain one feature vector $\boldsymbol{f}_{s,a}$ per node:

$$\mathcal{FWT}_s \boldsymbol{R} = [\boldsymbol{f}_{s,1}^\top | \boldsymbol{f}_{s,2}^\top | \cdots | \boldsymbol{f}_{s,N}^\top]^\top.$$

2. Estimate the distance matrix $\boldsymbol{D}_s(a, b)$:

$$\boldsymbol{D}_s(a, b) \simeq 1 - \hat{C}_{ab,\eta} = \frac{(\boldsymbol{f}_{s,a} - \bar{\boldsymbol{f}}_{s,a})^\top (\boldsymbol{f}_{s,b} - \bar{\boldsymbol{f}}_{s,b})}{||\boldsymbol{f}_{s,a} - \bar{\boldsymbol{f}}_{s,a}||_2 ||\boldsymbol{f}_{s,b} - \bar{\boldsymbol{f}}_{s,b}||_2}.$$

3. Same as step **3** of section IV.

Repeating these three steps for each scale $s \in \mathcal{S}$, one obtains the multiscale set of partitions $\mathcal{P} = \{P_s\}_{s \in \mathcal{S}}$.

VI. Detection of stable partitions

At this point of the discussion, we are able to output a set of partitions $\mathcal{P} = \{P_s\}_{s \in \mathcal{S}}$, one for each scale. The question one needs to address now can be formulated as a detection problem: how can one detect if a given scale displays relevant communities for a given graph?

The relevance of a scale is usually linked to notions of stability of the associated partition. In the literature, many notions of stability exist, e.g. some specific to multi-scale procedures [17] or some based on the stochasticity of modularity maximization algorithms [35], [36]. Only two different measures will be studied here, one from [17] and a new one that we introduce. The first measure relies on creating B resampled graphs by randomly adding $\pm p\%$ (typically $p = 10$) to the weight of each link and computing the corresponding B sets of partitions $\{P_s^b\}_{b\in[1,B],s\in\mathcal{S}}$. Then, for each scale s, define the stability $\gamma_r(s)$ as the mean of the similarity between all pairs of partitions of $\{P_s^b\}_{b\in[1,B]}$:

$$\gamma_r(s) = \frac{2}{B(B-1)} \sum_{(b,c)\in[1,B]^2, b\neq c} \mathtt{ari}(P_s^b, P_s^c), \qquad (19)$$

where the function $\mathtt{ari}$ is the Adjusted Rand Index, a partition similarity measure recalled in Appendix B (other similarities could be used indifferently). If, at a given scale s, the partition found for all B resampled graph is the same, the partition is stable ($\gamma_r(s)$ will be close to 1); if not, it is unstable ($\gamma_r(s)$ will be close to 0).

The second measure γ_a takes advantage of the inherent stochasticity of the fast community mining with random signals, presented here, as it is based on the transform of a few random signals.

Consider J sets of η random signals (typically J no larger than N/η to keep the computation time limited, the choice being $J = 20$ in the following) and compute the associated sets of partitions $\{P_s^j\}_{j\in[1,J],s\in\mathcal{S}}$. For each scale s, the stability $\gamma_a(s)$ is defined as the mean of the similarity between all pairs of partitions of $\{P_s^j\}_{j\in[1,J]}$:

$$\gamma_a(s) = \frac{2}{J(J-1)} \sum_{(i,j)\in[1,J]^2, i\neq j} \mathtt{ari}(P_s^i, P_s^j). \qquad (20)$$

Again, the more stable is the partition associated to the scale s, the closer to 1 will be $\gamma_a(s)$.

The two stabilities will be compared to each other in Section VIII-A. We argue that it is preferable to use γ_a than γ_r for several reasons. First, small scale structures are more sensitive than large scale structures for a same perturbation level p: there is a risk that small structures get artificially classified as unstable. Moreover, perturbing the graph perturbs also its spectrum, therefore the scale interval $[s_{min}, s_{max}]$ and ultimately the discrete set of scales $\mathcal{S}$. Therefore, each partition in $\{P_s^b\}_{b\in[1,B]}$ is not computed *exactly* at the same scale. Finally, γ_r requires to fix arbitrarily a parameter p. γ_a has none of these inconveniences. One could argue that the variance σ^2 is a parameter, but as we look at correlations, it actually does not have any impact. The only parameter one could find is η, but this parameter is not added by the stability measure: it is inherent to the community detection protocol.

VII. A STATISTICAL TEST FOR STABILITY

The protocol detailed in Section V outputs a set of partitions $\mathcal{P} = \{P_s\}_{s\in\mathcal{S}}$, and Section VI explains how to obtain a score γ_a that measures how stable each partition is. From this information, one can extract the K "best" scales of this network. Classical multiscale community mining methods stop at this point of the discussion: they output one partition per scale and a measure of their stability. The problem that arises next is that these methods *will* find the "best" K partitions of, for instance, an Erdös-Renyi (ER) graph, even though ER graphs have no community structure at any scale. In fact, what would be valuable is a way to inform us how *intrinsically* good each scale is. This issue exists for methods based on modularity [8], [17] for instance, as modularity maximisation outputs a solution even for ER graphs: what is the threshold value of modularity over which one decides that a given partition is objectively interesting?

In the framework of the proposed method, this turns into: what is the threshold value γ_a^{th} over which one may say a partition is sufficiently stable? We tackle this problem using stability statistics of randomised versions of the graph, against which the measured stabilities will be compared. Section VII-A presents a randomisation procedure using the Chung Lu (CL) model [37]. A statistical test is then defined in VII-B to automatically detect partitions that are statistically relevant. A general discussion on the test will be found at the end of the next Section (in VIII-C).

A. Randomised graphs

Consider a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \boldsymbol{A})$ and $(k_i)_{i=1,\dots,V}$ its degree sequence. A Chung-Lu (CL) graph [37], [38] associated to $\mathcal{G}$ is a binary random graph with the same expected degree sequence. To construct it, first randomly re-allocate all the degrees to the nodes, and wire each edge (connecting nodes i and j for instance) with a probability of $\min(1, \frac{k_i k_j}{2W})$ where $W = \frac{1}{2}\sum_i k_i$ is the expected total number of edges.

In applications where weighted graphs are considered (i.e., $\boldsymbol{A}$ is not binary), a model of weighted Chung-Lu graph [39] is recalled in Appendix C.

B. The test

Consider a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \boldsymbol{A})$ and its multi-scale set of partitions $\mathcal{P} = \{P_s\}_{s\in\mathcal{S}}$ and stability measure γ_a. To test which scale $s \in \mathcal{S}$ is interesting, we compare its stability measure to the stability measure of (weighted) Chung-Lu graphs. The details are:

1. Formulate a Null Hypothesis H0: $\mathcal{G}$ has no community structure at any scale.

2. Generate a large number R of randomised CL graphs associated to $\mathcal{G}$: $\{\mathcal{G}_1, \mathcal{G}_2, \cdots, \mathcal{G}_R\}$.

3. Compute the stability measure γ_a^r for each random graph $\mathcal{G}_r$, and empirically obtain the probability distribution $\mathcal{S}_{\gamma_a}$ from all the values $\{\gamma_a^r\}_{r\in[1,R]}$.

4. For each scale $s \in \mathcal{S}$, if $\gamma_a(s)$ is higher than the higher α-quantile γ_a^{th} of $\mathcal{S}_{\gamma_a}$, then we reject H0 with a confidence of $1 - 1/\alpha$ (if $R >> \alpha$): $\mathcal{G}$ has a community structure at this particular scale. Typically, we use $\alpha = 100$ and R large enough so that $\mathcal{S}_{\gamma_a}$ has a cardinal higher than 1000, i.e. $R \sim \frac{1000}{M}$. Indeed, each stability measure γ_a^r contributes (one value per scale) to $\mathcal{S}_{\gamma_a}$.

Finally, one ends up with a set of scales $\hat{\mathcal{S}} = \{s \in \mathcal{S} \text{ s.t. } \gamma_a(s) \geq \gamma_a^{\text{th}}\} \subset \mathcal{S}$ for which the associated partitions $\hat{\mathcal{P}} = \{P_k, k \in \hat{\mathcal{S}}\}$ are stable under significance level $1 - 1/\alpha$.

VIII. PERFORMANCE ON BENCHMARK NETWORKS AND COMPARISON TO OTHER METHODS

For a fair comparison of multiscale algorithms, we compare the set of partitions $\mathcal{P}$ found by each algorithm with the ground

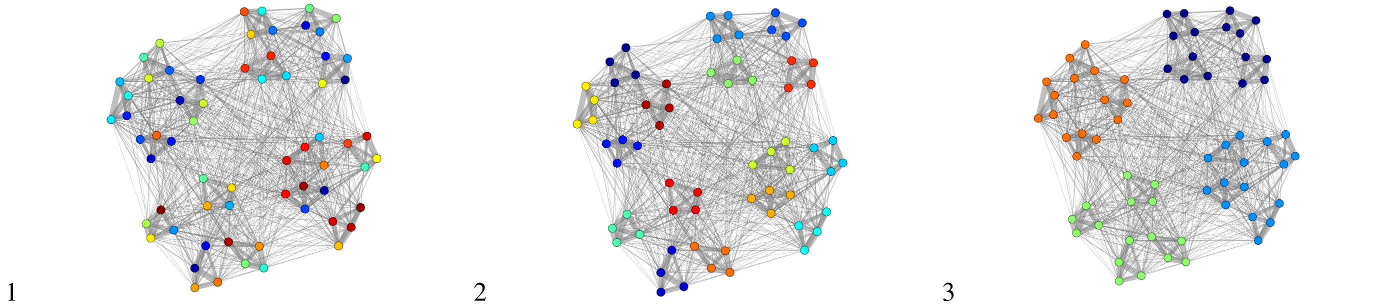


Fig. 3. Sketch of a realization of the SP graph. Each node displayed is in fact a community of 10 nodes. Three partitions (associated to the three stable intervals of scales of Fig. 4) are plotted in 1, 2 and 3, showing respectively the partition in 64, 16 and 4 communities (nodes drawn in the same color are in the same community). The layout of the graph is obtained using `ForceAtlas2` implemented in Gephi [40].

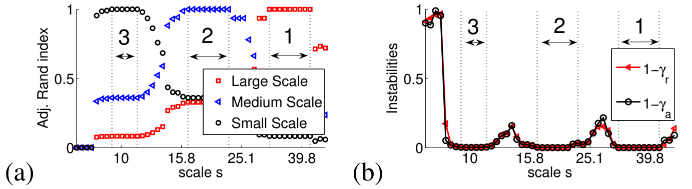


Fig. 4. (a) Result of the multiscale community mining protocol on a realization of a SP graph. Each scale outputs a partition and we plot its similarity with the small (medium, large) theoretical scale. The scale interval 1 (resp. 2, 3) represents the scales where the exact small (resp. medium, large) scale theoretical partition is uncovered. (b) Instabilities $1 - \gamma_a$ and $1 - \gamma_r$ versus scale s. The three intervals of scale are the same as in (a): the associated partitions of scales with low instability (i.e. high stability) correspond to the theoretical partitions.

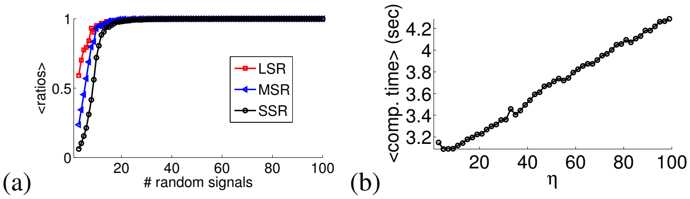


Fig. 5. (a) SSR, MSR and LSR with respect to η, the number of random vector used: only 30 random vectors are necessary to recover perfectly the three scales of description. (b) Computation time in seconds for the full community mining procedure with respect to η. Results are averaged over 100 realizations of a SP graph with $\rho = 1$ and $\bar{k} = 16$.

truth of well controlled graph benchmarks. The performance of a given algorithm is measured as the maximum value of the Adjusted Rand Index between the “true” partition of the benchmark and the partitions in $\mathcal{P}$. This measure has a name in information retrieval: the recall ratio. In our context of multiscale methods, we will use hierarchical benchmarks that have several ground truths. For instance, the first benchmark we use has three “true” partitions, corresponding to three different scales. We adapt the notion of recall ratio to this particular case: the large (resp. medium, small) scale recall LSR (resp. MSR, SSR) is the maximum value of the Adjusted Rand Index between the large (resp. medium, small) scale “true” partition and the partitions in $\mathcal{P}$.

A. Results on a Sales-Pardo network

For a first illustration of this method, we apply it on a three-level hierarchical graph benchmark first proposed by Sales-Pardo et al. [30], and recalled in Appendix A. A sketch of the three scales of this benchmark is illustrated in Fig. 3. Sales-Pardo (SP) graphs are parametrized by ρ that quantifies how separated the three scales are (the smaller is ρ the more separated the scales are), and $\bar{k}$, the average degree, that controls how dense the network is. The bigger is ρ and the smaller is $\bar{k}$ the harder it is to uncover the communities. We apply the costly protocol described in Section IV (which computes all $N = 640$ wavelets at each scale) to such a SP graph with parameters $\rho = 1$ and $\bar{k} = 16$: see Fig. 4a for an illustration of the result. There are intervals of scales where the theoretical partitions are exactly uncovered. In this particular graph, the recall ratios, i.e. the maximum value of the Adjusted Rand Index, are all equal to 1 (LSR=MSR=SSR=1).

To test the efficiency of the fast protocol based on random vectors described in Section V, to study the effect of η, and to compute γ_a, we applied our method with variable η to 100 realizations of the SP graph ($\rho = 1$, $\bar{k} = 16$). Fig. 5a shows that the true partitions are uncovered with a small number of random vectors ($\eta \simeq 30 << N = 640$) and that, as expected, more vectors are necessary to detect community structures at small scales. Here, 30 random vectors are enough to uncover all levels of description of the network, instead of the 640 wavelets. Fig. 5b shows the average computation time of the algorithm with respect to η: it is shorter than the 11.8 seconds required if one uses the 640 wavelets (computations for Matlab run on a laptop with Intel i7 Core@2.6GHz with 8GB of RAM). Results in terms of stability (γ_a, γ_r) are discussed in Section VIII-C.

Let us compare our method with two other methods from the multiscale community mining literature, namely Schaub et al.’s proposition [16] based on Markov processes on the graph and Arenas’ proposition [19] based on a parametrized modularity (and here optimized with Louvain’s algorithm [41]). We first compare these three methods on the SP benchmark with different sets of parameters. To this end, we used $\eta = 60$ random vectors for our method, the same number of scales for all methods ($M = 50$), and the scale boundaries proposed in the respective papers. Fig. 6a (resp. b and c) compares the LSR (resp. MSR and SSR) of the three methods, for $\rho = 1$ and different values of $\bar{k}$. Fig. 6d (resp. e and f) compares the LSR (resp. MSR and SSR) of the three methods, for the harder case $\rho = 2$ and different values of $\bar{k}$.

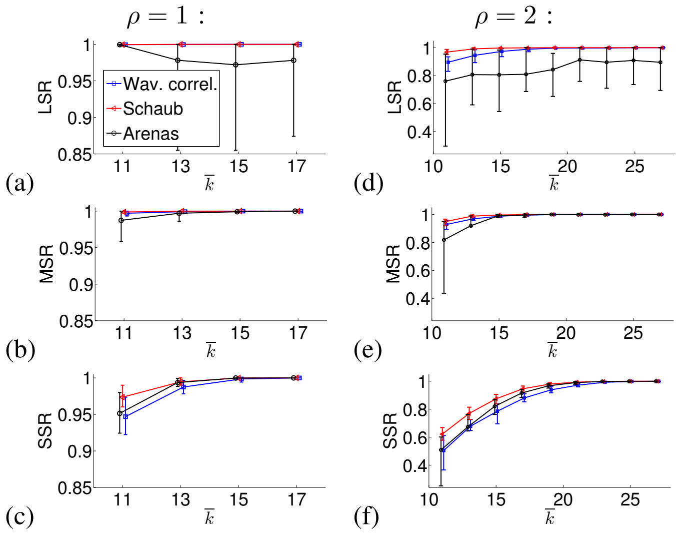


Fig. 6. Comparison between the LSR, MSR and SSR values obtained for three different multiscale community mining methods on the SP benchmark for different parameters: left (resp. right) column for $\rho = 1$ (resp. $\rho = 2$) and different values of $\bar{k}$. We plot the average and the 90% confidence intervals based on 20 realizations of SP graphs for each set of parameters.

B. Results on a LFR network

We also compare the three methods on the multiscale version of the Lancichinetti-Fortunato-Radicchi (LFR) benchmark [42], [43]. Codes to generate multiscale LFR graphs were retrieved from [44] and only non-overlapping graphs were created with the following set of parameters: $N = 300$ nodes with a mean degree of $k = 10$, a maximum degree of $kmax = 25$, a power law exponent of $t_1 = -2$ for the degree distribution, a power law exponent of $t_2 = -1$ for the community size distribution, a minimum of $minc = 10$ nodes and a maximum of $maxc = 50$ nodes for the micro community sizes, a minimum of $minC = 20$ and a maximum of $maxC = 80$ nodes for the macro community sizes, and no overlapping ($on = om = 0$). In this benchmark, there are only two community levels: a small scale level and a large scale level. Fig. 7a (resp. b) compares the LSR (resp. SSR) for a mixing parameter for the macro communities of $\mu_1 = 0.08$ and different values of inter-micro communities mixing values μ_2. Fig. 7c (resp. d) shows the same for $\mu_1 = 0.14$. To obtain this comparison, we used $\eta = 60$ random vectors for our method, the same number of scales for all methods ($M = 50$), and the scale boundaries proposed in the respective papers.

On average, our method does better than Arenas's proposition, and Schaub's method is slightly more accurate than ours. In terms of computation time, Arenas's version is quicker than the two others (that are comparable in time) as we used the fast Louvain algorithm to optimize their filtered modularity.

C. Results for the statistical test for stability

To illustrate the γ_a stability measure, let us use $J = 20$ sets of $\eta = 60$ random signals to estimate the stability γ_a of the SP graph used for Fig. 4a. The instability $1 - \gamma_a(s)$ is plotted in Fig. 4b: the three annotated intervals corresponding to intervals of scales where the theoretical partitions are exactly recovered correspond to high stability partitions (low instability $1 - \gamma_a$).

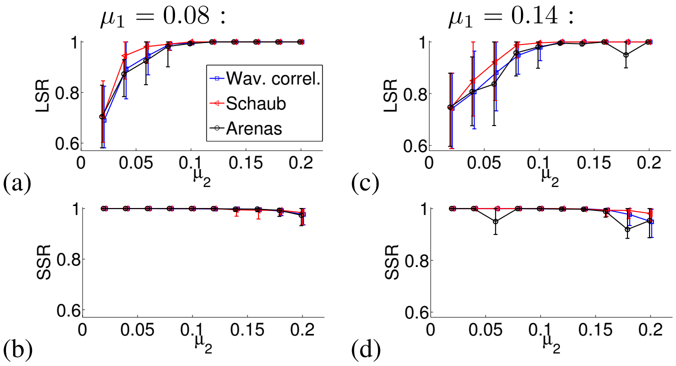


Fig. 7. Comparison between the LSR and SSR values obtained for three multiscale community mining methods on the LFR benchmark for different parameters: left (resp. right) column for $\mu_1 = 0.08$ (resp. $\mu_1 = 0.14$), and different values of μ_2. We plot the average and the 90% confidence intervals based on 20 realizations of LFR graphs for each set of parameters.

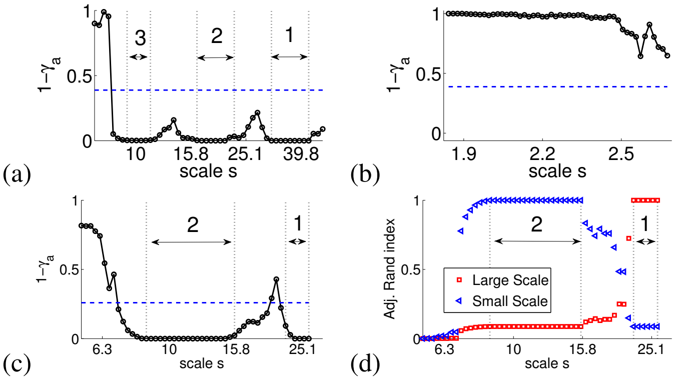


Fig. 8. Result of the stability test for (a) the SP graph studied in Fig. 4; (b) a randomised version (as explained in Section VII-A) of this SP graph; (c) a LFR graph with two scales of community structure as shown by the two intervals of scales where the exact theoretical partitions are uncovered [as shown in (d)]. The dashed horizontal line is the threshold value $1 - \gamma_a^{\text{th}}$ computed with the test. A scale with an instability $1 - \gamma_a$ (resp. stability γ_a) lower (resp. higher) than $1 - \gamma_a^{\text{th}}$ (resp. γ_a^{th}) is rejected by the test: its associated partition is more relevant than a typical partition found at that scale in a random graph.

The literature's instability $1 - \gamma_r(s)$ (computed with $B = 20$) is also plotted in Fig. 4b: both instabilities are here almost superimposed.

Up to our knowledge, there isn't any method in the multiscale community mining literature that we can compare our statistical test to. Therefore, we only illustrate its output on several examples grouped in Fig. 8: a) the SP graph studied in Fig. 4; b) a randomised version (as explained in Section VII-A) of this SP graph; c) a LFR graph with two scales of community structure as shown by the two intervals of scales where the exact theoretical partitions are uncovered (as shown in Fig. 8d). The ground truth in the case of Fig. 8a) is that at least the scales in the three intervals should have an instability lower than the threshold, which is the case. There are false positives, i.e. scales that have a lower instability than the threshold but for which the associated partitions are not one of the three theoretical ones. This is expected since the partitions corresponding to scales between the intervals are typically combinations of theoretical scales and are therefore more stable than partitions in random graphs. The SP graph

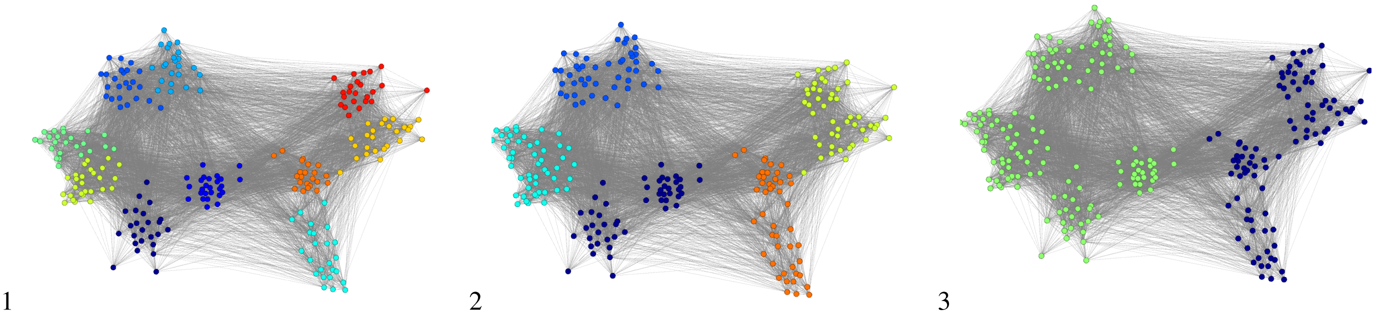


Fig. 9. Stable communities of a graph of social interactions between children in a primary school: figure 1 (resp. 2, 3) shows the partition in 10 (resp. 5, 2) communities (nodes drawn in the same color are in the same community). These 3 partitions are associated to the three stable scales of Fig. 10a. The layout of the graph is obtained using `ForceAtlas2` implemented in Gephi [40].

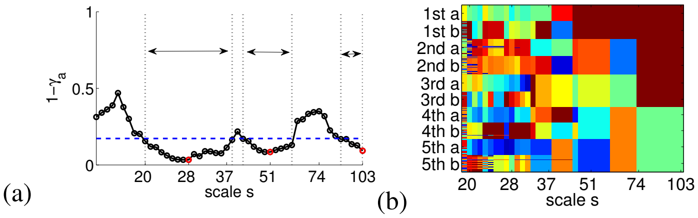


Fig. 10. (a) Results of the stability test for the social interaction network of Section IX-A. $\eta = 30$ random vectors were used. The dashed horizontal blue line is the stability threshold given by the statistical test: it separates three intervals of scales (represented by the vertical dotted lines). Each interval of scale can be represented by its highest stability scale (represented in red circles). (b) shows the corresponding partitions: the y-axis corresponds to the nodes ordered with respect to school grades. Two nodes in the same column have the same color if they are in the same community. The partitions corresponding to the eight first scales are not drawn here: they have too many communities for this mode of representation.

with these parameters is a particularly structured graph and partitions found at scales between the three intervals still show some stability. On a less structured graph like the LFR graph of Fig. 8c), there are less false positives. In the case of Fig. 8b), as the graph is random, the ground truth is that no partition at any scale is relevant: indeed, we find that no scale has an instability lower than the threshold. In the following, partitions that are not stable according to the test will be discarded, and when whole intervals of scales have partitions with an instability lower than the threshold, we select the significant local minima.

The issue of this statistical test is the computation time it requires. Indeed, one needs to compute the multiscale set of partitions of R different randomised graphs, where R is typically around 20. Therefore, it is not suitable for graphs larger than 1000 nodes.

IX. Three applications

We illustrate the method on three very different applications: a social network in IX-A, a non-uniformly sampled swiss roll manifold as an example of data that occurs in signal processing on networks in IX-B, and finally an example of a large SP graph in IX-C.

A. A graph of social interactions between children in a primary school

The method is first applied on a graph of social interactions between children in a primary school that was measured in 2009 by wearable RFID (Radio Frequency IDentification) sensors [45]. The aggregated data over two days is naturally represented by an adjacency weighted matrix $\boldsymbol{A}$ where $\boldsymbol{A}_{ij}$ represents the total time of contact between child i and child j. 242 children and teachers participated in the experiment, separated in five grades (from 1st grade to 5th grade), themselves separated in two classes per grade. The graph has $N = 242$ nodes and we use $M = 50$ scales.

Fig. 10 shows the results for this dataset. Three stable intervals of scales are uncovered (represented by the dotted vertical lines in Fig. 10a). Within each interval, we choose the scale with the highest stability (represented by a red circle in Fig. 10a) to be representative of the whole interval. Indeed, partitions within each interval are very similar: the mean similarity index between the representative partition of the small (resp. medium, large) scale interval and the other partitions of this interval is 0.93 (resp. 0.92, 1). Thereby, three scales of description stand out: the coarse scale ($s = 103$) where the older children (4th and 5th grades) are in one community and the younger ones (1st, 2nd and 3rd grades) in another. A medium scale description ($s = 51$) separates all the grades from one another (groups together all pairs of classes of same grade). The small scale ($s = 28$) separates all 10 classes from one another. These three partitions are also shown in Fig. 9.

B. The swiss roll manifold

A second example is based on the swiss roll manifold shown in Fig. 11. This manifold is created as in [20] except that the sampling points are non-uniformly sampled, drawing $N = 500$ points from 5 Gaussian distributions on the manifold. Then, a weighted graph is defined by using a Gaussian affinity kernel: $A_{ij} = \exp(-||x_i - x_j||^2/2\sigma^2)$, with $\sigma = 0.1$. We apply the method with $M = 50$ scales.

Fig. 12 shows the γ_a instability results for these data. Many scales are more stable than the statistical test's threshold. We choose to focus on the three most important local minima (represented by red circles in Fig. 12). Their associated partitions are plotted in Fig. 11: they separate the manifold in respectively 3, 5, and 13 communities. The scale parameter

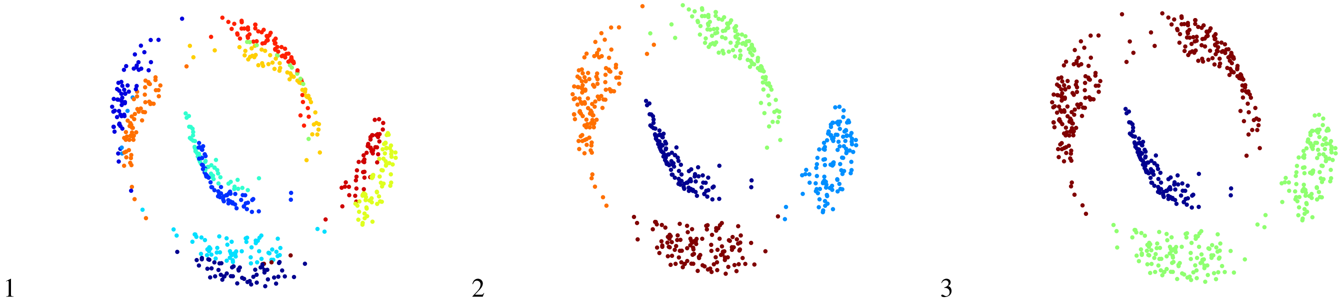


Fig. 11. Stable communities of a non uniform swiss roll manifold: Fig. 1 (resp. 2, 3) shows the partition in 13 (resp. 5, 3) communities (nodes drawn in the same color are in the same community). These 3 partitions are associated to the three stable scales of Fig. 12.

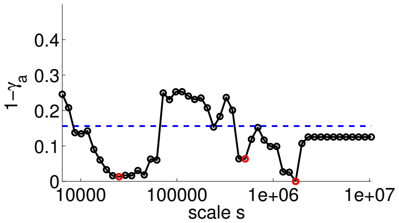


Fig. 12. Result of the stability test on the swiss roll manifold. $\eta = 50$ random vectors were used. The three most important local minima are drawn in red circles and their associated partitions are presented in Fig. 11.

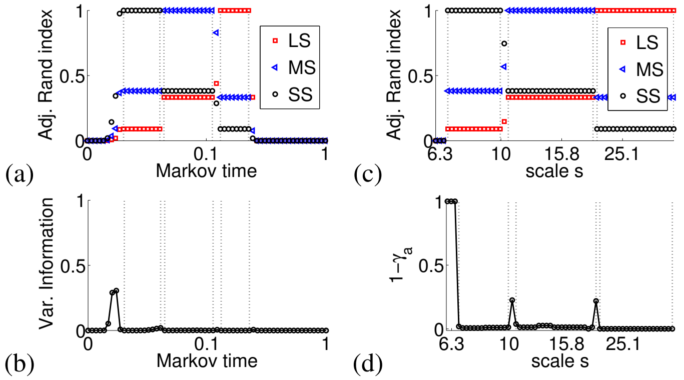


Fig. 13. Comparison between this paper's multi-scale community mining method [figures (c) and (d)] with Schaub et al.'s method [figures (a) and (b)] for a large SP graph of 6400 nodes. Both methods used $M = 60$ scales. For this paper's method, $J = 6$ sets of $\eta = 25$ random vectors were used.

s takes very high values because a few nodes of the graph are far from all others. Being almost disconnected, the graph Laplacian's first eigenvalues are very small, and s_{min} and s_{max} take therefore high values. This illustrates the importance of Section III in selecting automatically the relevant scale boundaries. Note that the intermediate scale perfectly recovers the five Gaussians used to generate the data, and that the smaller and larger scale partitions respect the geometry of the manifold.

C. A large Sales-Pardo graph

In order to show how the method behaves on larger graphs, a Sales-Pardo graph of $N = 6400$ nodes is considered with: $S_3 = 100$, $S_2 = 300$, $S_1 = 1200$, $S_0 = 6400 - 1600 = 4800$, $\rho = 1$ and $\bar{k} = 160$. In Fig. 13, we compare our method for $M = 60$ scales with Schaub et al. [16]'s method based on Markov dynamics. For Schaub's method, the instability measure used, at a given Markov time, is the variation of information between several solutions of the Louvain algorithm [41] at that Markov time (indeed, the Louvain algorithm is inherently stochastic). For this paper's method, we used here $\eta = 25$ random vectors and only $J = 6$ iterations to estimate γ_a but in this example, it is largely enough. In this case, the statistical test is not used as the computation cost would be prohibitive. Both methods have a similar average running time of 8 minutes to extract the multi-scale structure of this graph.

The result for the proposed method is promising for this larger graph, in that it points correctly to the three existing partitions at different scales, with an instability measure with sharp separations between them, sharper than the ones proposed by Schaub et al.'s method.

X. Conclusion

An original contribution to multiscale community mining in networks is discussed, relying on the recently defined spectral graph wavelets. The local information encoded in wavelets is used to probe node-to-node correlations depending on the scale. Then, a hierarchical clustering scheme finds the best partition in communities at each scale. We propose a way to by-pass the full computation of the wavelet correlation matrices by using the wavelet transform of a few random vectors, which improves the computational cost of the algorithm. Also, an original instability measure of partition in communities is introduced. This instability measure points at intervals of scales were the partitions appear to be relevant and stable. Along with a statistical test that compares the original graph to randomised ones, it enables us to output statistically significant scales at which communities exist –if they exist. The statistical test calls for improvements, as the randomisation procedure it uses destroys all communities, hence it cannot prevent us from falsely accepting combination of partitions at different scales as relevant ones. Still, the local minima of the instability curves (when lower than the threshold of the test) appear to clearly point to relevant partitions in communities.

The proposed general framework opens the way to new manners of dealing with complex network data and signals on them, by first aggregating the network using the proposed multiscale approach based on a notion of scale rooted in signal processing, before applying other techniques of analysis. We thus created a bridge between the emerging field of graph signal processing and its largest potential field of application : complex graph analysis. A future objective would be, for instance, to leverage the present work to define filtering operations on signals on graphs which would be consistent with the step of community detection in the graph.

SOFTWARE

A Matlab implementation of the Multi Scale Community Detection using graph WAVelets (MSCD_Wav) Toolbox is available online at [46].

APPENDIX A
THE SALES-PARDO (SP) HIERARCHICAL BENCHMARK

This hierarchical benchmark is a non-weighted graph introduced in [30], and later used in [17] to test multi scale community mining tools. We consider N nodes, and three community structures nested in one another: consider N/N_3 communities of N_3 nodes (the small scale level), nested in N/N_2 communities of N_2 nodes (the medium scale level), themselves nested in N/N_1 communities of N_1 nodes (the large scale level), where $N_3 < N_2 < N_1 < N$. Each node holds therefore 3 community memberships, one at each scale. Consider any node i and define S_x the number of nodes that hold x community memberships in common with i. Here:

- nodes that are not in i's large community do not hold any common community memberships with i: $S_0 = N - N_1$.
- nodes that are in i's large community but not in i's medium community only hold one common community membership with i: $S_1 = N_1 - N_2$.
- nodes that are in i's medium community but not in i's small community hold two common community memberships with i: $S_2 = N_2 - N_3$.
- nodes (different than i) that are in i's small community hold three common community memberships with i: $S_3 = N_3 - 1$.

Consider $\bar{k_3}$ the average (on the nodes) intra small-community degree, $\bar{k_2}$ the average intra medium-community (but extra small-community) degree, $\bar{k_1}$ the average intra large-community (but extra small and medium-community) degree and $\bar{k_0}$ the average extra large-community degree. We define:

$$\bar{k_0} = p_0 S_0, \qquad \bar{k_1} = p_1 S_1,$$
$$\bar{k_2} = p_2 S_2, \qquad \bar{k_3} = p_3 S_3, \tag{21}$$

where p_x is the probability of existence of a link between two nodes that hold x common memberships. A first parameter ρ tunes how well separated the different scales are:

$$\rho = \frac{\bar{k_0}}{\bar{k_1}} = \frac{\bar{k_0} + \bar{k_1}}{\bar{k_2}} = \frac{\bar{k_0} + \bar{k_1} + \bar{k_2}}{\bar{k_3}}. \tag{22}$$

The smaller is ρ the more separated are the scales, the easier it is to extract the hierarchical community structure. A second parameter, the average degree $\bar{k}$, controls how dense the network is:

$$\bar{k} = \bar{k_0} + \bar{k_1} + \bar{k_2} + \bar{k_3}. \tag{23}$$

The smaller is $\bar{k}$, the sparser is the graph, the harder it is to recover the communities. Given a pair of parameters $(\rho, \bar{k})$, we obtain the following equations for the probabilities p_i:

$$p_0 = \frac{\rho^3}{(1+\rho)^3}\frac{\bar{k}}{S_0}, \qquad p_1 = \frac{\rho^2}{(1+\rho)^3}\frac{\bar{k}}{S_1},$$
$$p_2 = \frac{\rho}{(1+\rho)^2}\frac{\bar{k}}{S_2}, \qquad p_3 = \frac{\bar{k}}{(1+\rho)S_3}. \tag{24}$$

The p_i being probabilities between 0 and 1, we have an implicit constraint:

$$\frac{\bar{k}}{1+\rho} \leq S_3. \tag{25}$$

In this paper, we consider $N = 640$ nodes, and three community structures nested in one another: 64 communities of $N_3 = 10$ nodes (the small scale level), nested in 16 communities of $N_2 = 40$ nodes (the medium scale level), themselves nested in 4 communities of $N_1 = 160$ nodes (the large scale level). Therefore, $S_0 = 480$, $S_1 = 120$, $S_2 = 30$ and $S_3 = 9$. Then, given the parameters ρ and $\bar{k}$ one chooses to consider, apply equation (24) to get the probabilities of link existence to generate the graph. In IX-C, we use larger N_1, N_2, N_3 with $N = 6400$.

APPENDIX B
THE ADJUSTED RAND INDEX OF SIMILARITY

Let P and P' be two partitions we want to compare, and:

- a be the number of pairs of nodes that are in the same community in P *and* in the same community in P'.
- b be the number of pairs of nodes that are in different communities in P *and* in different communities in P'.
- c be the number of pairs of nodes that are in the same community in P *and* in different communities in P'.
- d be the number of pairs of nodes that are in different communities in P *and* in the same community in P'.

In other words, $a+b$ is the number of "agreements" between P and P', and $c+d$ is the number of "disagreements" between P and P'. The Rand index is given by:

$$R = \frac{a+b}{a+b+c+d} = \frac{a+b}{\binom{n}{2}}. \tag{26}$$

The Adjusted Rand (AR) index is the corrected-for-chance version of the Rand index:

$$\texttt{simi}(P, P') = \frac{R - ExpectedIndex}{MaxIndex - ExpectedIndex}, \tag{27}$$

as explained in details in [47]. For instance, this corrects the fact that two partitions in two communities have a higher chance to have a high Rand Index than two partitions in twenty communities. The choice of this similarity index is not crucial; another one could be used with no loss of generality of the method discussed here.

APPENDIX C
WEIGHTED CHUNG-LU GRAPHS

In many applications, the adjacency matrix A is weighted, and for the test of Section VII-B, one needs a weighted version of the classical Chung-Lu model: we present here weighted

Chung-Lu graphs (wCL). To this aim, we first create a CL graph, and then allocate a weight to each edge. In real networks, weights and topology are often not independent [48]. In fact, there is often a correlation between the average strength of nodes and their degree (we recall that the strength of a node is the sum of the weights of its edges). In order to keep these correlations, we compute from $\mathcal{G}_0$ the empirical distribution $P_k(w)$ of the weights of the links attached to nodes of degree k[1].

A wCL graph associated to $\mathcal{G}_0$ is then built in the following way: start by creating a CL graph with the same expected degree sequence as $\mathcal{G}_0$. For each node i (of degree k_i) of this CL graph, draw weights from the appropriate distribution P_{k_i} and randomly allocate them to its links whose weight has not yet been specified (if i is linked to a node j that has already been considered, then the weight of link $i-j$ has already been chosen by using P_{k_j} and is not computed again).

We thereby obtain a wCL graph with the same expected degree sequence as $\mathcal{G}_0$, the same strength-degree correlation and a similar weight sequence than $\mathcal{G}_0$ [39]. A wCL graph associated to a binary graph $\mathcal{G}_0$ is a CL graph.

Finally, depending on the data at hand, other models could be considered to obtain randomised graphs for the statistical test of Section VII-B.

Acknowledgments

The authors would like to thank the three anonymous reviewers for their constructive comments, which greatly helped to improve the paper, Ronan Hamon for fruitful discussions, and Alain Barrat along with the SocioPatterns collaboration for providing access to the primary school contact data.

References

[1] N. Tremblay and P. Borgnat, "Multiscale community mining in networks using the graph wavelet transform of random vectors," in *Global Conference on Signal and Information Processing (GlobalSIP), 2013 IEEE*, Dec 2013, pp. 463–466.

[2] ——, "Partitionnement multi-échelle d'un graphe en communautés : détection des échelles pertinentes," in *GRETSI*, Brest, France, 2013.

[3] ——, "Multiscale detection of stable communities using wavelets on networks," in *European Conference on Complex Systems*, Barcelona, Spain, 2013.

[4] ——, "Multiscale community mining in networks using spectral graph wavelets," in *Signal Processing Conference (EUSIPCO), 2013 Proceedings of the 21st European*, Marrakech, Morocco, Sept 2013.

[5] "Special issue : Adaptation and learning over complex networks," *Signal Processing Magazine*, vol. 30, no. 3, 2013.

[6] S. Fortunato, "Community detection in graphs," *Physics Reports*, vol. 486, no. 3-5, pp. 75–174, 2010.

[7] A. Bertrand and M. Moonen, "Seeing the bigger picture," *Signal Processing Magazine*, vol. 30, no. 3, pp. 71–82, 2013.

[8] M. Newman, "Modularity and community structure in networks," *PNAS*, vol. 103, no. 23, p. 8577, 2006.

[9] S. S. Tabatabaei, M. Coates, and M. G. Rabbat, "Ganc: Greedy agglomerative normalized cut," *arXiv preprint*, 2011. [Online]. Available: arxiv.org/abs/1105.0974

[10] M. Rosvall and C. T. Bergstrom, "Maps of random walks on complex networks reveal community structure," *PNAS*, vol. 105, no. 4, pp. 1118–1123, 2008.

[11] J. Richiardi, S. Achard, H. Bunke, and D. V. D. Ville, "Machine learning with brain graphs," *Sig. Proc. Mag.*, vol. 30, no. 3, pp. 58–70, 2013.

[12] R. F. Betzel, A. Griffa, A. Avena-Koenigsberger, J. Goni, J.-P. Thiran, P. Hagmann, and O. Sporns, "Multi-scale community organization of the human structural connectome and its relationship with resting-state functional connectivity," *arXiv preprint*, 2013. [Online]. Available: arxiv.org/abs/1304.0485

[13] E. Ravasz, A. L. Somera, D. A. Mongru, Z. N. Oltvai, and A.-L. Barabasi, "Hierarchical organization of modularity in metabolic networks," *Science*, vol. 297, no. 5586, pp. 1551–1555, 2002.

[14] S. Fortunato and M. Barthelemy, "Resolution limit in community detection," *PNAS*, vol. 104, no. 1, p. 36, 2007.

[15] 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.

[16] 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.

[17] R. Lambiotte, "Multi-scale modularity in complex networks," in *Proc. 8th Int. Symp. WiOpt*, 2010, pp. 546–553.

[18] J. Reichardt and S. Bornholdt, "Statistical mechanics of community detection," *Phys. Rev. E*, vol. 74, no. 1, p. 016110, 2006.

[19] 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.

[20] 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.

[21] M. Crovella and E. Kolaczyk, "Graph wavelets for spatial traffic analysis," in *INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications. IEEE Societies*, vol. 3, 2003.

[22] R. Coifman and M. Maggioni, "Diffusion wavelets," *Applied and Computational Harmonic Analysis*, vol. 21, no. 1, p. 5394, 2006.

[23] R. Tibshirani, G. Walther, and T. Hastie, "Estimating the number of clusters in a data set via the gap statistic," *Journal of the Royal Statistical Society: Series B*, vol. 63, no. 2, pp. 411–423, 2001.

[24] F. R. Chung, *Spectral graph theory*. American Mathematical Society, Providence, USA., 1997, vol. 92.

[25] N. Leonardi and D. Van de Ville, "Tight wavelet frames on multislice graphs," *IEEE Transactions on Signal Processing*, vol. 61, no. 13, pp. 3357–3367, 2013.

[26] A. Sandryhaila and J. Moura, "Discrete signal processing on graphs," *IEEE Trans. on Signal Processing*, vol. 361, no. 7, pp. 1644–1656, 2013.

[27] L. Donetti and M. Muñoz, "Improved spectral algorithm for the detection of network communities," *Arxiv preprint*, 2005. [Online]. Available: arXiv.org/abs/physics/0504059

[28] D. Shuman, P. Vandergheynst, and P. Frossard, "Chebyshev polynomial approximation for distributed signal processing," in *DCOSS, 2011 International Conference on*, 2011, pp. 1–8.

[29] U. Von Luxburg, "A tutorial on spectral clustering," *Statistics and computing*, vol. 17, no. 4, pp. 395–416, 2007.

[30] 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.

[31] M. Mitrovic and B. Tadic, "Spectral and dynamical properties in classes of sparse networks with mesoscopic inhomogeneities," *Phys. Rev. E*, vol. 80, p. 026123, Aug 2009.

[32] A. Jain, M. Murty, and P. Flynn, "Data clustering: a review," *ACM computing surveys (CSUR)*, vol. 31, no. 3, pp. 264–323, 1999.

[33] B. King, "Step-wise clustering procedures," *Journal of the American Statistical Association*, vol. 62, no. 317, pp. 86–101, 1967.

[34] T. Hastie, R. Tibshirani, J. Friedman *et al.*, *The elements of statistical learning: data mining, inference, and prediction*. Springer New York, 2001.

[35] M. Seifi and J.-L. Guillaume, "Community cores in evolving networks," in *21st WWW conference*. ACM, 2012, pp. 1173–1180.

[36] T. Chakraborty, S. Srinivasan, N. Ganguly, S. Bhowmick, and A. Mukherjee, "Constant communities in complex networks," *Scientific reports*, vol. 3, 2013. [Online]. Available: http://dx.doi.org/10.1038/srep01825

[37] F. Chung and L. Lu, "The average distances in random graphs with given expected degrees," *PNAS*, vol. 99, no. 25, pp. 15 879–15 882, 2002.

[38] J. Miller and A. Hagberg, "Efficient generation of networks with given expected degrees," in *Algorithms and Models for the Web Graph*, ser. Lecture Notes in Computer Science, A. Frieze, P. Horn, and P. Praat, Eds. Springer Berlin Heidelberg, 2011, vol. 6732, pp. 115–126.

[1] If there are not enough nodes of degree k in the original data to obtain a reasonable fit, we use the 50 nodes whose degrees are closest to k to compute $P_k(w)$.

[39] N. Tremblay, A. Barrat, C. Forest, M. Nornberg, J.-F. Pinton, and P. Borgnat, "Bootstrapping under constraint for the assessment of group behavior in human contact networks," *Phys. Rev. E*, vol. 88, p. 052812, Nov 2013.

[40] M. Bastian, S. Heymann, M. Jacomy *et al.*, "Gephi: an open source software for exploring and manipulating networks." *ICWSM*, vol. 8, pp. 361–362, 2009.

[41] V. Blondel, J. Guillaume, R. Lambiotte, and E. Lefebvre, "Fast unfolding of communities in large networks," *Journal of Statistical Mechanics: Theory and Experiment*, vol. 2008, no. 10, p. P10008, 2008.

[42] A. Lancichinetti, S. Fortunato, and F. Radicchi, "Benchmark graphs for testing community detection algorithms," *Phys. Rev. E*, vol. 78, p. 046110, Oct 2008.

[43] A. Lancichinetti and S. Fortunato, "Benchmarks for testing community detection algorithms on directed and weighted graphs with overlapping communities," *Phys. Rev. E*, vol. 80, p. 016118, Jul 2009.

[44] https://sites.google.com/site/andrealancichinetti/files/.

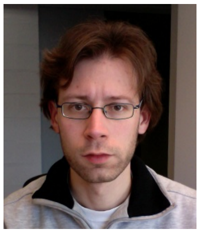
[45] J. Stehle, N. Voirin, A. Barrat, C. Cattuto, L. Isella, J. Pinton, M. Quaggiotto, W. Van Den Broeck, C. Regis, B. Lina, and P. Vanhems, "High-resolution measurements of face-to-face contact patterns in a primary school," *PloS one*, vol. 6, no. 8, p. e23176, 2011.

[46] http://perso.ens-lyon.fr/nicolas.tremblay/index.php?page=downloads.

[47] L. Hubert and P. Arabie, "Comparing partitions," *Journal of Classification*, vol. 2, no. 1, pp. 193–218, 1985.

[48] A. Barrat, M. Barthélemy, R. Pastor-Satorras, and A. Vespignani, "The architecture of complex weighted networks," *PNAS*, vol. 101, pp. 3747–3752, 2004.

Nicolas Tremblay received in 2009 the M.Sc. degree in theoretical Physics with a minor in complex systems, from the École Normale Supérieure de Lyon (ENS), France. He currently works towards a Ph.D. degree in Signal Processing applied to complex networks at the Laboratoire de Physique, ÉNS de Lyon. His research interests include graph analysis, graph signal processing and their applications in social or biological networks.



Pierre Borgnat is Researcher at CNRS in Signal Processing, at the Laboratoire de Physique, ENS de Lyon, France. He was born in France in 1974, was received at the École Normale Supérieure de Lyon in 1994 and as a Professeur Agrg in Physical Sciences in 1997. He got a Ph.D. degree in physics and signal processing in 2002. He worked in 2003-2004 in the Signal and Image Processing group of the IRS, IST (Lisbon, Portugal). Since October 2004, he has been a full-time CNRS researcher. His research interests are in statistical signal processing of non-stationary processes (time-frequency, time warping, test of stationarity, EMD,...), of scaling phenomena (time-scale, wavelets) and in processing of graph signals and complex networks. He works on several applications of these signal processing methods, for instance in Internet traffic modeling and measurements, and their applications (for traffic classification, anomaly identification,...), for fluid mechanics, for analysis of social data, or for transportation studies.