



HAL
open science

Scale-space module detection for random fields observed on a graph non embedded in a metric space

Bernard Chalmond

► **To cite this version:**

Bernard Chalmond. Scale-space module detection for random fields observed on a graph non embedded in a metric space. 2014. hal-00947472v2

HAL Id: hal-00947472

<https://hal.science/hal-00947472v2>

Preprint submitted on 27 Oct 2014

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Scale-space module detection for random fields observed on a graph non embedded in a metric space

Bernard Chalmond ^{*†}

October 2014

Abstract

In the spirit of Lindeberg's approach for image analysis on regular lattice, we adapt from a statistical viewpoint, the blob detection procedure for graphs non embedded in a metric space. We treat data observed on such a graph in the goal of detecting salient modules. This task consists in seeking subgraphs whose activity is strong or weak compared to those of their neighbors. This is performed by analyzing nodes activity at multi-scale levels. To do that, data are seen as the occurrence of a univariate random field, for which we propose a multi-scale graphical modeling. In the framework of diffusion processes, the covariance matrix of the random field is decomposed into a weighted sum of graph Laplacians at different scales. Under the assumption of Gaussian law, the maximum likelihood estimation of the weights is performed that provides a set of relevant scales. As a result, we obtain a multi-scale decomposition of the random field on which the module detection is based. This method is experimentally analyzed on simulated data and biological networks.

Keywords. Blob Detection, Module Detection, Network Activity, Graphical Modeling, Scale-space Random Field, Graph Laplacian, Diffusion Kernel, Multi-scale Decomposition, Scale Selection

*CMLA, UMR CNRS 8536, ENS Cachan, France

†Cergy-Pontoise University, France

1 Introduction

This paper is related to the following general issue : given an undirected graph $G = (V, E)$ with only one component non embedded in a metric space, and an observation \mathbf{x} of a univariate real random field \mathbf{X} indexed by the nodes V of this graph, one seeks subgraphs $\{\mathcal{M}_k\}$ in V for which the respective observations $\{x_{\mathcal{M}_k}\}$ appear as salient profiles, in comparison to their surrounding. Such a subgraph with its profile $(\mathcal{M}_k, x_{\mathcal{M}_k})$ is called module. In summary, we have the following schema :

$$[\{x_i\}_{i \in V}, G = (V, E)] \rightsquigarrow \{(\mathcal{M}_k, x_{\mathcal{M}_k})\}, \quad (1)$$

where $x_i \in \mathbf{R}$. This concept depends on the context and what we seek. We use the term module in a broad sense. However, we focus in the following on a particular module called blob or spot, depending of the context.

Concretely, the problem is as follows. Consider Fig. 3 that shows an image $\{x_i, i \in V\}$ where V denotes the nodes of a sampling grid \mathcal{L} included in \mathbf{R}^2 . This image shows a multitude of dark spots of various sizes that a scale-space algorithm has detected. This detection uses the graph $\mathcal{L} = (V, E)$ where E are the nearest neighbor connections. Here, our visual perception clearly distinguishes the spots. While keeping the values $\{x_i\}$, suppose now that we replace \mathcal{L} by a graph G whose nodes are not in a metric space. Displaying \mathbf{x} requires to represent the graph in the plane, which implies to choose a particular layout. This is illustrated in Fig. 7 and Fig. 8(c) that display an "image" on such a graph with three different layouts. This image contains five distinct spots that it is difficult to recognize, although these figures display the same image \mathbf{x} . Similarly, in the case of Fig. 3, our perception of spots would be greatly disturbed, and the detection algorithm would not work because it requires a metric space. Dealing with this problem is the subject of our article. To our knowledge it has not been still treated.

Although different, this problem suggests another problem that needs to be presented in order to avoid some confusions. Let a set of points $\{\vec{\xi}_i, i \in V\}$ in space \mathbf{R}^p with $p > 1$, allowing to define a similarity matrix W between these points, e.g. from their correlations or distances. This matrix is then used to infer a connectivity structure E , typically by connecting highly correlated or spatially close nodes. The detection of modules is then performed on the graph $G = (V, E)$, for example by using the concept of betweenness [19]. Here, the objective is the determination of $\{\mathcal{M}_k\}$. In summary, we have the following two-step schema :

$$\begin{aligned} \text{(i)} \quad & \{\vec{\xi}_i\}_{i \in V} \rightsquigarrow W \rightsquigarrow E, \\ \text{(ii)} \quad & G = (V, E) \rightsquigarrow \{\mathcal{M}_k\}. \end{aligned} \quad (2)$$

This second schema is further discussed in Section 1.1. In a nutshell, we can say that (2) seeks sub-networks in G whereas (1) seeks *active* sub-networks with respect to \mathbf{x} .

The tackled issue is resulting from molecular biology for which a vast literature exists. With respect to our concerns, a few references are [17, 9, 35, 27, 11, 22]. This

issue arises also in other fields, like social networks where modules refer to communities [1, 26]. One crucial step when studying the structure and dynamics of these networks is to identify modules/communities. However, these studies are mainly devoted to the schema (2) for biological networks and (2-ii) for social networks, whereas we are interested by the schema (1) due to the nature of our data, which come from a univariate random field.

The connectivity of G is summarized in the graph Laplacian matrix L , which plays a central role in our context :

$$L_{i,j} = \begin{cases} -1 & \text{if } i \sim j \\ d_i & \text{if } i = j \\ 0 & \text{otherwise,} \end{cases} \quad (3)$$

where $i \sim j$ denotes the edge $(i, j) \in E$, and $d_i = \sum_j \mathbf{1}_{j \sim i}$ is the degree of node i , i.e. the number of edges connected to i . L is a symmetric positive semi-definite matrix, which can be written as

$$L = D - A$$

where $D = \text{diag}\{d_i\}$ and A is the binary adjacency matrix $a_{i,j} = 1$ if $i \sim j$. Note that the graph Laplacian appears when one considers the local variation energy, called bending energy :

$$U(\mathbf{x}) = \sum_{i \sim j} (x_i - x_j)^2 = \mathbf{x}' L \mathbf{x}, \quad (4)$$

where the sum is over the edges $(i, j) \in E$. In (3) each edge $i \sim j$ carries the value $a_{i,j} = 1$. This definition can be extended to the weighted case, where the nonnegative weights are not necessarily all equal to 1. In both cases, we have $d_i = \sum_j a_{i,j}$.

We continue this introduction by positioning our contribution with respect to previous works. Then, the model and the methodology are presented in Section 2. The multi-scale decomposition and the module detection are tested on simulated random fields and on real data in Section 3, where diagnostic tools are introduced. We invite the reader to have a look at Fig. 8 that illustrates and summarizes the method.

1.1 Graph partitioning

Although module detection is not a partitioning task, some aspects of the related problem of spectral partitioning could lead to confusions. There is a large literature on spectral clustering for graph partitioning [36, 30, 28, 10, 3] among many others. In spectral clustering [30, 36], given a graph as in (2-ii) we compute eigenvectors u_1, \dots, u_m associated to the m smallest eigenvalues μ_1, \dots, μ_m of L , and assign to every node i the vector $\{u_k(i)\}_{k=1}^m$ in \mathbf{R}^m . Then, graph partitioning is the outcome of a vector space clustering algorithm such as k -means applied to the resulting vectors. In background of this procedure, there is an important property. If the graph is composed with c connected components, then the first c eigenvalues of L are zero, and the corresponding eigenvectors are the indicator vectors of the connected components. Fig.5 illustrates such a graph with 4 components.

This approach also works when one replaces the adjacent/ nonadjacent coefficients $a_{i,j}$ by a similarity or closeness measure : $a_{i,j} = w(i,j)$. The multiplicity of the eigenvalue 0 is the number of connected components of the underlying graph where nodes i, j are adjacent when $w(i,j) > 0$.

Two examples illustrate the closeness measure for the schema (2). When the graph Laplacian represents a 3D discrete surface (mesh), every node $i \in V$ is associated with a 3D coordinate point $\vec{\xi}_i$ in \mathbf{R}^3 , also denoted \vec{v}_i [31]. The weight of an edge $i \sim j$ is defined by the Gaussian function $w(i,j) = \exp(-(\|\vec{v}_i - \vec{v}_j\|^2/\sigma_v^2))$. Hence, the geometric structure of the mesh is encoded in the weights. The second example concerns graph based image segmentation [28]. The image is $\{x_i, i \in V\}$ where V are the nodes of a 2D regular grid embedded in \mathbf{R}^2 . Every node $i \in V$ is associated with a 3D vector $\vec{\xi}_i = (\vec{v}_i, x_i)$ where \vec{v}_i is a 2D coordinate point. The Gaussian weight function is rewritten $w(i,j) = \exp(-(\|\vec{v}_i - \vec{v}_j\|^2/\sigma_v^2 + |x_i - x_j|^2/\sigma_x^2))$. Other weighting functions were proposed in the literature.

Two pixels are connected if they are within distance δ : $w(i,j) \neq 0$ if $\|\vec{v}_i - \vec{v}_j\| < \delta$. But how to chose the graph connection radius δ ? In [10], from heuristic considerations, the graph weights are segmented into different scales :

$$W = W_1 + W_2 + \dots + W_r, \quad (5)$$

where W_s corresponds to a specific spatial separation range : $w_s(i,j) \neq 0$ if $\delta_{s-1} < \|\vec{v}_i - \vec{v}_j\| < \delta_s$.

In our case, the graph G is not embedded in an Euclidean space as the mesh in the examples above. Although non uniform weights $a_{i,j}$ can be chosen, these weights are not necessarily associated to a distance. To perform blob extraction, we use the diffusion property based on the graph Laplacian that does not require to have an explicit closeness measure. Since diffusion is a multi-scale process, we take advantage of this property to define a decomposition of the affinity matrix. This decomposition is related to generalized additive models that bring a theoretical base [37, 14].

1.2 Blob detection

On a regular mesh

In the image analysis domain, when G is simply a regular grid embedded in \mathbf{R}^2 , the problem of salient area detection has received much attention and in particular for blob detection [24], as illustrated in Fig.3. In this figure, the detected blobs are localized by squares whose size (scale) is adapted to the width of the blobs.

A blob is regarded as a spot and a simple model is given by the Gaussian profile [29]. In this case, we have the following result on which the scale-space blob detection is based. Consider the simple image $\mathbf{x} = \{x_i, i \in V\}$ representing a Gaussian spot characterized by a width parameter λ_0 and centered at a point \vec{v}_0 on the grid : $x_i \propto \exp(-\|\vec{v}_i - \vec{v}_0\|^2/2\lambda_0)$, for every $i \in V$. Consider a smooth version $\tilde{\mathbf{x}}^{(\lambda)}$ of the image

obtained by convolution with Gaussian kernel \mathcal{G}_λ :

$$\begin{aligned}\tilde{\mathbf{x}}_i^{(\lambda)} &= \sum_{i'} x_{i'} \mathcal{G}_\lambda(i, i') = \mathcal{G}_\lambda(i, \cdot) \star \mathbf{x}, \\ \mathcal{G}_\lambda(i, i') &\propto \exp(-(\|\vec{v}_i - \vec{v}_{i'}\|^2/2\lambda)).\end{aligned}\tag{6}$$

[24] gives the following property that the spot center satisfies :

$$\left. \frac{d}{d\lambda} \lambda [\Delta \tilde{\mathbf{x}}^{(\lambda)}]_{i_0} \right|_{\lambda=\lambda_0} = 0,\tag{7}$$

where Δ denotes the discretized Laplacian operator on the grid. In the image processing literature, $\Delta \mathcal{G}_\lambda$ called Laplacian of Gaussian, is used for multi-resolution representations [33]. Laplacians of Gaussian have mathematical properties, which have been widely studied in the scale-space community. (7) tell us that the derivative of $\lambda \Delta \mathcal{G}_\lambda$ is able to select the width λ_0 of the Gaussian spot. Essentially, $\lambda \Delta$ quantifies in some sense a curvature of the smoothed spot, and this curvature is optimal when $\lambda = \lambda_0$. This property is used to detect blobs in the images : the detected blob centers are the local extrema of the discretized scale-space volume

$$\{\lambda [\Delta \tilde{\mathbf{x}}^{(\lambda)}]_i, i \in V, \lambda \in \Lambda\},\tag{8}$$

where Λ is a finite set of scales corresponding to an increasingly coarse sub-sampling of the regular grid. For every detected blob, the optimization of (8) returns a scale, which characterizes the width of the blob.

From mesh to graph non embedded in a metric space

In this paper, module refers to the extension of the blob concept to graphs. In a first step, this extension is straightforward since (6) is the solution of the heat equation on \mathbf{Z}^2 whose extension to graphs is well known [20].

However, extending the blob detection to non geometric graphs requires some modifications with respect to scale and space. While for a mesh, it is natural to choose Λ from a sub-sampling of the grid ([12], Chap. 10), for non geometric graph this choice is much less trivial since the relevant scales are irregularly spread in \mathbf{R}^+ , and the scale has no explicit dimension. In this goal, the multi-scale representation $\{\tilde{\mathbf{x}}^{(\lambda)}, \lambda \in \Lambda\}$ must be revisited in order to get a sparse representation denoted $\{\mathbf{x}^{(\lambda)}, \lambda \in \Lambda\}$ yielding a non-redundant decomposition of \mathbf{x} in term of reconstruction : $\sum_\lambda \mathbf{x}^{(\lambda)} = \mathbf{x}$, a property that $\tilde{\mathbf{x}}^{(\lambda)}$ does not satisfy. This property of non-redundancy is necessary for the identification of the right scales.

1.3 Module, semantic module and related works

In image analysis, the module detection is used first for extracting areas of interest without using any strong prior information. These areas are then interpreted with greater precision or extended using high-level information in order to obtain semantic modules, as for

object recognition [8]. This remark holds also in systemic biology where semantic modules correspond to biological modules (see [17, 9] among many others). The definition of biological modules does not rely solely on areas and profiles, but also uses complex biological knowledge. In several papers, the detection of biological modules operates in two stages: firstly, detection of module seeds, or more simply modules, and secondly refinement of the detected modules to finally obtain meaningful biological modules [35].

We comment some main approaches for module detection as introduced in the bioinformatic literature. Given a scoring function that allows to compute the importance of every sub-network, finding the maximal-scoring connected subgraph is an NP-problem. In the seminal work [17], the main limitation is that node scores are treated independently since the sub-network score is calculated as a sum of the node scores. To overcome this limitation, [9] proposes an inverse problem approach in which the node scores are modeled by a hidden Markov random field model under a constraint of regularity that is expressed by a bending energy as (4). Two major well-known drawbacks are inherent to this approach [5]: the data-driven determination of the regularity scale (the trade-off parameter), and the energy minimization that requires stochastic optimization, a difficult computation task, already encountered in [17]. But conceptually, the main limitation of the Markovian model is that it is mono-scale, which is not suitable when the size of the modules is varying.

Instead of using the bending energy at a single scale, we propose to use it with a multi-scale formulation in order to adapt the scale to the module sizes. Technically, the advantage of this approach is twofold. First, the set of relevant scales can be estimated efficiently from the data. Second, we avoid the huge computation burden of the stochastic optimization. The computation is limited to scan a multi-scale representation of type (6) by searching the differential local extrema as it is done for blob detection on a grid \mathcal{L} .

2 Models and Method

2.1 Random Field and Diffusion Process

This section summarizes a set of fundamental results on graph Laplacian and diffusion kernels. Consider a random field $\mathbf{X} = (X_1, \dots, X_n)'$ observed on an undirected graph $G = (V, E)$. V denotes the node set and E the edges connecting them. The dependency structure between the random variables $\{X_i\}$ depends on the topological structure given by E . This dependency structure is here limited to a covariance structure modeled by a diffusion kernel [25], a choice explored in many domains and especially in pattern recognition, biological networks analysis and image processing [2, 32, 39].

We seek to represent \mathbf{X} by a random field model on G , denoted $\mathbf{Y}(\lambda)$, whose covariance structure depends on a scale parameter $\lambda > 0$. Essentially, this model is obtained by equalizing the variations due to a change of scale, with the spatial variations as follows :

$$Y_i(\lambda + d\lambda) - Y_i(\lambda) \doteq d\lambda \sum_{j \in V: j \sim i} (Y_j(\lambda) - Y_i(\lambda)), \quad (9)$$

and in vector form :

$$\begin{aligned} \mathbf{Y}(\lambda + d\lambda) - \mathbf{Y}(\lambda) &= -d\lambda L \mathbf{Y}(\lambda), \\ L &= D - A, \end{aligned} \quad (10)$$

where the graph Laplacian L is defined in (3). The equation (10) is the discretized version on G of the classical heat differential equation : ¹

$$\begin{cases} \frac{d}{d\lambda} \mathbf{Y}(\lambda) &= -L \mathbf{Y}(\lambda), \\ \mathbf{Y}(0) &= \mathbf{X}. \end{cases} \quad (11)$$

whose solution is

$$\mathbf{Y}(\lambda) = K_\lambda \mathbf{X}. \quad (12)$$

$$K_\lambda = e^{-\lambda L}, \quad (13)$$

K_λ is a matrix exponential whose definition is $e^M = \sum_{i=0}^{\infty} \frac{M^i}{i!}$. For every node, one has :

$$Y_i(\lambda) = \sum_{j: j \sim i} K_\lambda(i, j) X_j = K_\lambda(i, \cdot) \mathbf{X}, \quad (14)$$

which is the generalization of (6). The exponential of a symmetric matrix providing a semi-definite positive matrix, the matrix K_λ , which is called diffusion kernel, can be used as a covariance matrix for modeling the covariance between the random variables $\{X_i\}$. The more λ is large, the more the off-diagonal effects in K_λ increase. λ is interpreted as a scale parameter and $Y_i(\lambda)$ as a scale-space random field on $V \times \mathbf{R}^+$.

By nature, the diffusion kernel has a multi-scale property that is well identified, and especially for dimensionality reduction applications [23]. However, the choice of its scale parameter λ remains a difficulty [11]. For small λ , $K_\lambda(i, i)$ reflects local properties of G around the node i , while for large λ it captures some global structures. For instance, in the geometry processing field, the diagonal term $K_\lambda(i, i)$ has been used as a shape descriptor [31] by considering that for every λ , the local spatial extrema of this function provide a feature-based scale-space representation of shapes, useful for shape matching.

2.2 Graphical Modeling

The outstanding issue at the end of the previous modeling step is the choice of λ . In other words what is the scale λ the most representative of the observed profile \mathbf{x} . In fact, several scales may explain this profile. Therefore, a natural approach consists of decomposing \mathbf{X} into r independent random fields according to a discrete set of relevant scales $\Lambda = \{\lambda_1 < \dots < \lambda_r\}$:

$$\mathbf{X} = \sum_{j=1}^r \mathbf{X}^{(j)} + \mathbf{X}^{(0)}, \quad (15)$$

¹In the classic case of diffusion in \mathbf{R}^2 , λ is a time parameter.

where $\mathbf{X}^{(j)}$ denotes the random field at scale λ_j and $\mathbf{X}^{(0)}$ a residual [16, 34]. Following the idea of Fourier decomposition, for every profile \mathbf{x} , the $\{\mathbf{x}^{(j)}\}_{j=1}^r$ can be seen as frequency components of \mathbf{x} , from high to low frequencies. The decomposition (15) is related to the additive spline models whose theoretical foundation can be traced back to [37] Chap.10, (see also [14]) and later reintroduced under the name of multiple kernel in the machine learning community [21]. Note that $\mathbf{X}^{(j)}$ does not match $\mathbf{Y}(\lambda_j)$ in (12), since the sum $\sum_j \mathbf{Y}(\lambda_j)$ over a given set of scales does not reconstruct \mathbf{X} .

In our approach, the covariance matrix $\text{Cov}(\mathbf{X}^{(j)})$ of every component is modeled from the diffusion kernel (13). So we use r kernels $\{K_{\lambda_1}, \dots, K_{\lambda_r}\}$ denoted $\{K_1, \dots, K_r\}$, such that $\text{Cov}(\mathbf{X}^{(j)}) = K_j = \sigma_j^2 \kappa_j$ where κ_j is given by (13) at scale λ_j . Due to the independence of the components, the covariance matrix $\text{Cov}(\mathbf{X})$ is the following multi-scale diffusion kernel :

$$\begin{aligned} \bar{K}_{\sigma, \Lambda} &= \sum_{j=0}^r K_j = \sum_{j=0}^r \sigma_j^2 \kappa_j \\ &= \sum_{j=1}^r \sigma_j^2 e^{-\lambda_j L} + \sigma_0^2 I_n . \end{aligned} \tag{16}$$

Each kernel κ_j is weighted by a positive parameter σ_j^2 that is all the more great than the scale λ_j significantly contributes to the random field \mathbf{X} . The covariance matrix K_0 is that of a white noise. As we said above, the more λ_j is large, the more the off-diagonal effects in K_j increase. In other words, when λ_j increases, the components $\{\mathbf{X}^{(j)}\}$ are increasingly smooth. The passage from $\mathbf{X}^{(j)}$ to $\mathbf{X}^{(j+1)}$ implies that some details in $\mathbf{X}^{(j)}$ are attenuated.

If we assume that the dependency structure of the random variables $\{X_i\}$ is uniquely described by its kernel, then it is legitimate to consider that \mathbf{X} is distributed according to the Gaussian law

$$\mathbf{X} \sim \mathcal{N}(0, \bar{K}_{\sigma, \Lambda}) . \tag{17}$$

The scales $\{\lambda_j\}_{j=0}^r$ and their associated weights $\sigma_j^2 = \{\sigma_j^2\}_{j=0}^r$ are unknown parameters that are estimated using the maximum likelihood principle².

Although the theoretical mean of \mathbf{X} is zero, the empirical mean of each observed sub-profile $x_{\mathcal{M}_k}$ is not necessarily zero, as for instance in Fig. 9. This is due to high scales that create long range correlations, or in other words low frequencies. Understanding the diffusion kernel is not a trivial task, this requires to call the graph spectral theory [25]. Note also that the choice of the diffusion kernel as covariance matrix arises as a necessity because we have only one observation of \mathbf{X} . If we could have many observations then the covariance matrix could be estimated.

In comparison with the heuristic decomposition (5), which uses a hard multi-scale separation of the weights, the multi-scale representation (16) appears as a soft decomposition based on the overall structure of the graph via L , and moreover allows statistical estimation of each component contribution.

²For notational convenience, we introduce $\lambda_0 = 0$ that is associated to K_0 .

• *Weight estimation.* For a given Λ , let $\ell(\sigma|\Lambda) = \log(p_{\sigma,\Lambda}(\mathbf{x}))$ be the log-likelihood of σ , where $p_{\sigma,\Lambda}$ denotes the probability density of \mathbf{x} . Given an observation \mathbf{x} and the Gaussian $\mathcal{N}(0, \bar{K}_{\sigma,\Lambda})$, the log-likelihood is

$$\ell(\sigma|\Lambda) = -\log |\bar{K}_{\sigma,\Lambda}| - \mathbf{x}' \bar{K}_{\sigma,\Lambda}^{-1} \mathbf{x} + Cte, \quad (18)$$

where Cte denotes a constant term. The maximum likelihood estimate is computed under the constraint of positivity of the parameters σ :

$$\hat{\sigma}(\lambda) = \underset{\sigma}{\operatorname{argmax}} \ell(\sigma|\Lambda) \text{ under the constraint } \sigma > 0. \quad (19)$$

For moderate sizes of n , the non-linear programming algorithms using gradient descent techniques are operational. For larger dimensions, the computation of the determinant $|\bar{K}_{\sigma,\Lambda}|$ and the inverse $\bar{K}_{\sigma,\Lambda}^{-1}$ becomes more difficult [18]. To reduce the amount of computation, one might also wonder whether it would be possible to remove the term $\log |\bar{K}_{\sigma,\Lambda}|$ in the likelihood, in order to work only with the generalized least-squares $\mathbf{x}' \bar{K}_{\sigma,\Lambda}^{-1} \mathbf{x}$. Theoretically, we know that this estimate is not statistically consistent [13]. Our experiments have confirmed this defect, by showing severe aberrations in the multi-scale decompositions (cf. Section 3.1).

• *Scale estimation.* A procedure for selecting the set Λ is now required. Given a uniform discretization Λ_0 of the scale domain in \mathbf{R} , the scale selection procedure estimates a subset Λ of scales irregularly distributed in Λ_0 , which explains the profile of \mathbf{x} according to a given criteria :

$$\Lambda^0 = \{\lambda_j^0 = j\delta, j = 1, \dots, r^0\} \rightsquigarrow \Lambda = \{\lambda_{j_1}^0, \dots, \lambda_{j_r}^0\},$$

where δ is the discretization stepsize. First, the estimation $\hat{\sigma}(\Lambda^0)$ is computed according to (19). To determine r , we perform a diagonalization of the covariance matrix

$$\tilde{K} = \sum_{j=1}^{r^0} \hat{\sigma}_j^2(\Lambda^0) e^{-\lambda_j^0 L}, \quad (20)$$

of which we retain only the r largest eigenvalues $\nu_1 \geq \dots \geq \nu_r$ according to the criteria

$$\frac{\sum_{i=1}^r \nu_i}{\sum_{i=1}^{r^0} \nu_i} = 1 - \epsilon, \quad (21)$$

where ϵ is a positive parameter chosen close to 0, typically $\epsilon = 0.01$ or 0.025 . This criterion is related to that used in Principal Component Analysis [15]. It means the dispersion of \mathbf{X} can be approximatively represented by r linearly independent components with an information loss determined by ϵ . Finally, from the estimated r , we can then achieve the selection of relevant scales Λ . These scales are associated with the r largest $\hat{\sigma}_j^2(\Lambda^0)$, i.e. the scales whose components are the most involved in the dispersion of \mathbf{X} . These scales are denoted $\{\lambda_{j_1}^0, \dots, \lambda_{j_r}^0\}$ or more simply $\{\lambda_1, \dots, \lambda_r\}$. As a consequence of (21), the

estimates $\hat{\sigma}_j^2$ associated to the scales in $\Lambda^0 \setminus \Lambda$ are much lower than those in Λ , and even close to 0. This selection achieves a pruning of non-significant scales.

• *Statistical Multi-scale Decomposition.* This task concerns the estimation of the r components $\mathbf{X}^{(j)}$ of the multi-scale decomposition of \mathbf{X} . Component estimation is closely linked to the scale selection problem. Denote $\tilde{K}U = UD_\nu$ the spectral equation of the previous diagonalization, where D_ν is the diagonal matrix of the r largest eigenvalues Λ of \tilde{K} . Because of (21), assume that the eigenvalues $\lambda_0 \setminus \lambda$ are approximately equal to zero. This is especially true when ϵ is very small. Beyond r_0 , the eigenvalues $\nu_{r_0+1} \geq \dots \geq \nu_n$ are smaller and we can consider they are all close to 0. In this case, one can write $\tilde{\mathbf{X}} = U\mathbf{B}$ where \mathbf{B} are the coordinates of $\tilde{\mathbf{X}}$ on the eigenvectors U . To estimate the scale components, we take into account the importance of each eigenvalue, and this, using a Bayesian estimation with a prior distribution related on these eigenvalues.

Proposition 1 *Given an observation \mathbf{x} and the prior distribution $\mathbf{B} \sim N(0, D_\nu)$ the Bayesian estimation provides the scale components :*

$$\hat{\mathbf{x}}^{(j)} = K_j U D_\nu^{-1} \hat{\mathbf{b}}, \quad \forall j = 1, \dots, r \quad (22)$$

$$\hat{\mathbf{b}} = (\sigma_0^2 D_\nu^{-1} + I_r)^{-1} U' \mathbf{x}. \quad (23)$$

The proof is given in Appendix 5.1. In this proof, if we replace the spectral equation relative to Λ by the equation relative to Λ_0 : $\tilde{K}U^0 = U^0 D_\nu^0$, then Proposition 1 is still valid. This is useful when we do not assume that all eigenvalues $\Lambda_0 \setminus \Lambda$ are negligible. In this case, the Bayesian estimation is more justified because of the high difference between the values of Λ and $\Lambda_0 \setminus \Lambda$.

2.3 Module Detection

Given an undirected graph G and an observation \mathbf{x} of the random field, we first compute the estimated scales $\{\lambda_j, j = 1, \dots, r\}$ and the associated decomposition (22) $\{\mathbf{x}^{(j)}, j = 1, \dots, r\}$ as presented in the previous section. Rather than considering directly the components, we consider their spatial variations with respect to the graph Laplacian L :

$$\begin{aligned} (L\mathbf{x}^{(j)})_v &= \sum_{i: i \sim v} (x_v^{(j)} - x_i^{(j)}), \quad \forall v \in V \\ &= d_v x_v^{(j)} - \sum_{i: i \sim v} x_i^{(j)}. \end{aligned} \quad (24)$$

This specifies the regularity of each component. $L\mathbf{x}^{(j)}$ is all the more great positively (resp. negatively) than the expression $x_v^{(j)}$ of the node v is strongly increasing (resp. decreasing) with respect to its neighbors. Therefore we look for nodes that are most differentially expressed with respect to L , and this by examining the expression of the components at different scales. Since the amplitude of variations of $L\tilde{\mathbf{x}}^{(j)}$ decreases when

the scale increases, a specific normalization is required. As in the case of blob detection (8) on a lattice \mathcal{L} , an efficient normalization is $\lambda_j L\tilde{\mathbf{x}}^{(j)}$. A scale λ_j for which $\lambda_j(L\mathbf{x}^{(j)})_v$ is a local extremum with respect to scale and space, is seen as reflecting a module at position v and scale λ_j . This implies the following procedure.

For any node $v \in V$, we denote by $N_v^k \subset V$ the relative nodes of v of order k , ($k = 1, \dots, \kappa$). $k = 1$ means the nearest neighbors (NN), $k = 2$ means the nearest neighbors of v to which their NN are added, etc... The module detection consists in searching local optima of the components with respect to the neighborhoods as follows³ :

$$\begin{aligned} \forall v \in V : (j(v), k(v), v'(v)) = \underset{j, k, v' \in N_v^k}{\operatorname{argopt}} \lambda_j(L\mathbf{x}^{(j)})_{v'} ; \\ \text{if } v'(v) = v, \text{ then } v \text{ is a module center at scale } \lambda_{j(v)} . \end{aligned} \quad (25)$$

When a module center is detected at v , its area \mathcal{M}_v is defined by the subgraph $\{v, N_v^{k(v)}\}$. In the next section, we denote $\overset{\circ}{V}$ the nodes corresponding to the detected module centers, and therefore the set of detected areas is written as :

$$\{\mathcal{M}_v, v \in \overset{\circ}{V}\} . \quad (26)$$

3 Experiments

Recall that a module is an active subgraph denoted $x_{\mathcal{M}}$ where \mathcal{M} is a subgraph of G . A regular lattices \mathcal{L} do not show particular structure like stars or clusters, unlike the case of irregular graphs G . Let us give an example of graph showing a particular structure. The graph structure is organized around known subgraphs $\{R_k = (V_k, E_k)\}$ called regulons (or hubs). A regulon is a set of nodes $V_k \subset V$ connected to one or several common nodes, called regulators. A regulon can be connected to several regulators and a regulator can be connected to several regulons. Fig.5 shows a graph with four regulons and four regulators.

In practice, such regulons can be used a posteriori for interpreting the detected modules or inferring semantic modules. Depending on the profile of \mathbf{x} , the area \mathcal{M} of a module can be simply a regulon, a subregulon or the union of several regulons. Fig.6 shows a short time-series $\{\mathbf{x}(1), \mathbf{x}(2), \mathbf{x}(3)\}$ of a random field \mathbf{X} observed on the graph of Fig.5. The colors depict the output of the scale-space module detection performed on every $\mathbf{x}(t)$. Successively, 3, 3 and 2 modules were detected, while the graph is composed of 4 regulons.

3.1 Evaluation on Simulated Data

3.1.1 Simulation Procedure

For phenomena of high complexity, simulated data are an important preliminary support for modeling when we do not have data with sufficient knowledge of the "ground truth".

³ $v'(v)$ also depends on k , what is omitted to simplify the writing.

The simulation of the random field \mathbf{X} requires to give the ground truth, consisting of a graph $G = (V, E)$ and the parameters (Λ, σ) . In our procedure, G is organized in regulons : $G = \uplus_{k=1}^m R_k$. Here, we assume for simplicity that each regulon R_k is associated to only one regulator r_k . The symbol $+$ in \uplus indicates that the regulons are mutually connected. This high-level of connection is equivalent to a graph G^B between the regulators : $G^B = (\{r_k\}, E_r)$.

The simulation is done in two steps. First, the simulation of a graph G consisting of m regulons is done as described in Appendix 5.2. Second, a sample \mathbf{x} of $\mathbf{X} \sim \mathcal{N}(0, \bar{K}_{\Lambda, \sigma})$ is drawn. To do that, we simply simulate $\boldsymbol{\alpha} \sim \mathcal{N}(0, I_n)$ since the diagonalization $\bar{K}_{\Lambda, \sigma} = VD V'$ implies $VD^{1/2}\boldsymbol{\alpha} \sim \mathcal{N}(0, \bar{K}_{\Lambda, \sigma})$.

3.1.2 Simulated Data

Fig.8(a) shows the inter-regulon graph G^B and Fig.8(b) the graph G . Each regulon has its own color. Fig.8(c) displays an observation \mathbf{x} of the random field \mathbf{X} on G , and Fig.9 shows its 1-D profile. In this experiment, each regulon is a potential module since the simulation procedure is based on a regulon structure. \mathbf{x} was simulated using the multi-scale kernel (16) with 3 scales : $\lambda_1 = 8$, $\lambda_2 = 14$, $\lambda_3 = 24$ and $\sigma_1 = \sigma_2 = \sigma_3 = 1$.

Although the theoretical mean of \mathbf{X} is zero, the mean of each observed regulon in Fig.9 is not zero. However, due to the correlation between regulators, two regulons may have similar mean levels. This situation is favorable to the concept of module. This is consistent with Fig. 6 wherein there are 3 detected modules for 4 regulons. Note that in the absence of observation \mathbf{x} , the spectral partitioning as recalled in Introduction, detects 4 modules corresponding to the 4 regulons.

3.1.3 Data Analysis

The estimation and detection tasks are illustrated in the figures 8, 9, 10 and 11. The maximum likelihood estimation (19) was performed using the scale domain $\Lambda^0 = \{2k, k = 0, 1, \dots, 15\}$. Fig.10 displays the statistical multi-scale decomposition. The continuous black line connecting the data points is the sum $\sum_{k=1}^{15} \hat{\mathbf{x}}^{(2k)}$ of all the components except the noise component $\hat{\mathbf{x}}^{(0)}$. Since this line interpolates the data points, this means that the estimated noise component is very low. The selected Λ is computed from this decomposition using (21) with $\epsilon = 0.1$. The three main components associated to $\{\lambda_1 = 8, \lambda_2 = 10, \lambda_3 = 30\}$ are shown in Fig.9. It is interesting to compare this statistical decomposition with the ordinary scale-space decomposition (12) shown in Fig.11. The statistical decomposition has the ability to focus more clearly on the spectral content of \mathbf{x} . $\lambda_3 = 30$ reflects low frequencies whereas $\lambda_2 = 10$ contributes to high frequencies. But above all, it remedies the redundancy of ordinary scale-space representation, and therefore favors the identification of the right scales. The detected modules shown in Fig.8(d) correspond to the rule (25). In Fig.9, the locations of the detected extrema are indicated by red circles. There is exactly one detected module per regulon.

This procedure is statistically assessed by Monte Carlo simulation. The random field \mathbf{X} is simulated 200 times under the same conditions as above. From the obtained samples

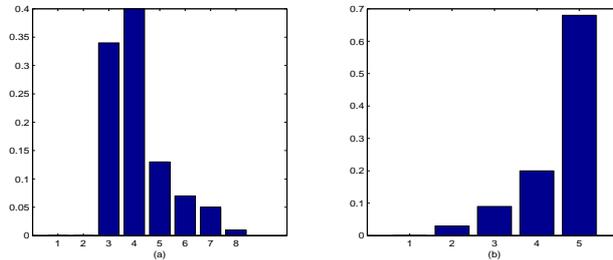


Figure 1: (a) Histogram of the empirical probabilities $\{\hat{P}(|\Lambda| = k)\}_{k=1}^8$ of the number of selected scales, (b) Histogram of the empirical probabilities $\{\hat{P}(|\hat{V}| = k)\}_{k=1}^5$ of the number of detected modules.

$\{\mathbf{x}(\ell), \ell = 1, \dots, 200\}$, the probability of the number of selected scales $\{P(|\Lambda| = k)\}_{k=1}^8$ were estimated for $\epsilon = 0.01$, as well as the probability of the number of detected module centers $\{P(|\hat{V}| = k)\}_{k=1}^5$, as shown in Fig. 1. The number of detected modules is random, with a main mode at $\hat{V} = 5$. In fact, as noted above, the mean levels of two regulons may be substantially close and therefore be recognized as belonging to the same module if they are connected.

Above, we have mentioned the prominent role of the term $\log |\bar{K}_{\sigma, \Lambda}|$ in the likelihood. This is confirmed experimentally. Without this term, the estimation-detection procedure was repeated on the same data as previously. The results are shown in Fig.12. The multi-scale decomposition is then quite inaccurate. All weights are very high and the scale components are close to zero.

3.2 Bacillus Subtilis Data

Fig.4 illustrates the multi-scale decomposition of a field \mathbf{x} that represents gene expressions of Bacillus Subtilis. The underlying graph $G = (V, E)$ comes from the regulatory network of the bacterium. V denotes genes, E connections between genes and \mathbf{x} gene expressions on V (Fig.4-a)⁴. The entire graph contains 1607 genes, 2345 edges and 132 regulons. Fig.5 displays four connected regulons extracted from this network. In steady-state, gene expressions are assumed to be governed by the model (17). In Fig.4 we see the structuring effects of the method in terms of gene grouping as this had already been shown for other regulatory networks [11].

In many applications, one is interested in studying the change of modules across different conditions [22]. In our example, the expression of the regulons depends on the nutritional environment of the bacteria over time, some of them are over-expressed and

⁴The biological network has been simplified by removing the protein level network and therefore in G the regulatory protein-encoding genes and their proteins are confused. Furthermore, the edge directions in E have been deleted. Consequently, we cannot speak strictly of regulation in the sense of regulatory networks. .

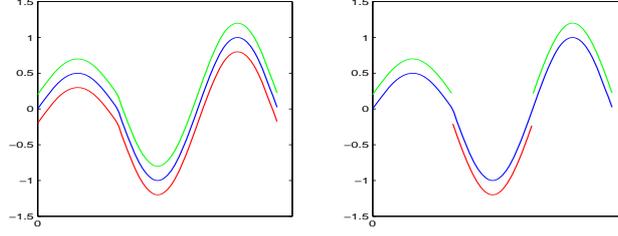


Figure 2: A toy example with three active regulons : two over-expressed and one under-expressed. (a) $\hat{\mu}$ in blue. Confidence band : $\{\mu_{R_k}^+\}_{k=1}^3$ in green, $\{\mu_{R_k}^-\}_{k=1}^3$ in red. (b) A particular configuration $\mathbf{x}'(\ell) = \{\mu_{R_1}^+, \mu_{R_2}^-, \mu_{R_3}^+\}$.

other ones are under-expressed. With the module detection, we search to identify regions of the graph that are particularly expressed through time. Fig.6 illustrates this detection on a short time series of the random field observed at 3 time points. The detection has been done at every time t , independently of the others. Every detected module is composed of one or several regulons. For instance, at time $t = 1$, there are 3 detected modules, which are respectively depicted in green, yellow and pink. The green module is composed of two regulons, what can favor the semantic interpretation of the module from properties of the regulons.

The validation is primarily based on biological aspects. In the considered experience, one examines nutrient change effects : an experimental population of cells grows first in Glucose and then Malate is injected at time t_0 such that $1 \leq t_0 \leq 11$. The detected modules should reflect this change. A further biological analysis is beyond the scope of our article, [4, 7]. However, a diagnostic tool is now proposed to help with this analysis. The idea is to generate configurations of \mathbf{X} using a confidence band around the obtained decomposition

$$\hat{\mu} = \sum_{j \in \Lambda} \hat{\mathbf{x}}^{(j)},$$

in order to quantify the stability of the detected modules. We start with the bootstrap confidence interval described in [38] that we recall. As a result of the decomposition, the estimated residuals are : $\hat{\mathbf{x}}^{(0)} = \mathbf{x} - \hat{\mu}$, whose empirical variance is $\hat{\sigma}_0^2 = \|\hat{\mathbf{x}}^{(0)}\|^2/n$. Derived from (15), a generative model based on $\hat{\mu}$ is written as :

$$\mathbf{X} = \hat{\mu} + \mathbf{X}^{(0)},$$

where $\mathbf{X}_i^{(0)}$ are independent Gaussian random variables $LG(0, \hat{\sigma}_0^2)$. Generate $\mathbf{X}^{(0)}$ allows to simulate \mathbf{X} , which is now distributed according to $\mathcal{N}(\hat{\mu}, \hat{\sigma}_0^2 I_n)$. Pretending that $\hat{\mu}$ is the "true" μ , generate N bootstrap samples $\{\mathbf{x}(\ell)\}_{\ell=1}^N$, and compute their respective smooth profiles $\{\hat{\mu}(\ell)\}_{\ell=1}^N$. Using these samples, for every node $i \in V$, and for a given confidence

γ close to one, a confidence interval of μ_i denoted $[\mu_i^-, \mu_i^+]$ is estimated, which provide a confidence band $[\mu^-, \mu^+]$ as detailed in [38].

Our validation uses this confidence band to generate configurations. Denote $\{\mu_{R_k}^-\}$ and $\{\mu_{R_k}^+\}$ the two confidence profiles viewed from the regulons. For generating a new configuration \mathbf{x}' , we draw randomly for each regulon R_k between $\mu_{R_k}^-$ and $\mu_{R_k}^+$, as illustrated in Fig. 2. Repeating this process N' times, we obtain new samples $\{\mathbf{x}'(\ell)\}_{\ell=1}^{N'}$, on which module detection is performed. Finally, among the N' detected fields $\{\{\mathcal{M}_v(\ell), v \in \overset{\circ}{V}(\ell)\}, \ell = 1 \dots N'\}$, we compute the proportion of fields that fit with the field $\{\mathcal{M}_v, v \in \overset{\circ}{V}\}$ obtained on the original \mathbf{x} , cf. (26). This proportion is associated to the confidence γ , providing a quantitative diagnostic tool.

The experiments show that module detection puts into light the activated modules and therefore provides a mean to study dynamic random fields. However, module detection on time series has been performed without taking into account time dependence. At every time t , the observation of the random field has been treated independently of the others. Nevertheless, it is well known that Markovian dependence can improve the sensitivity of the detection of isolated low signal. In the related paper [6], we present a Markovian spatio-temporal modeling that generalizes the present model. Doing so, in Fig.1 the probability $\hat{P}(|\overset{\circ}{V}| = 5)$ should be higher.

4 Conclusion

This paper proposes and implements a multi-scale graphical modeling for univariate random vectors observed on an undirected graph. The result is a multi-scale decomposition of the random field which provides an analysis tool to deal with specific treatments because it allows to select relevant scales. This tool is especially used for module detection. With hindsight, this detector seems relatively simple. However, emphasis has been put on a coherent modeling without heuristics and with very few tunable parameters.

5 Appendix

5.1 Proof of Proposition 1

Let $\tilde{\mathbf{X}} = \mathbf{X} - \mathbf{X}^{(0)} = \sum_{j=1}^{r^0} \mathbf{X}^{(j)}$, and recall the spectral equation $\tilde{K}U = UD_\nu$ where $\tilde{K} = \sum_{j=1}^{r^0} K_j$. First, since the columns of U are independent, we can write $\tilde{\mathbf{X}} = U\mathbf{B}$ where \mathbf{B} is a r -random vector. Then, the spectral equation allows to rewrite

$$\begin{aligned} \tilde{\mathbf{X}} = U\mathbf{B} = \tilde{K}UD_\nu^{-1}\mathbf{B} &= \sum_{j=1}^{r^0} K_jUD_\nu^{-1}\mathbf{B} = \sum_{j=1}^{r^0} \mathbf{X}^{(j)}, \\ \text{where } \mathbf{X}^{(j)} &= K_jUD_\nu^{-1}\mathbf{B}, \end{aligned} \quad (27)$$

which provides the components (22). Second, $\tilde{\mathbf{X}} = U\mathbf{B}$ implies the covariance matrix

$$\text{Cov}(\mathbf{B}) = U\tilde{K}U' = D_\nu . \quad (28)$$

For a given observation \mathbf{x} , the Bayesian estimation of the occurrence of \mathbf{B} consists in maximizing the log-likelihood $\log p(\mathbf{b} | \mathbf{x}) = \log p(\mathbf{x} | \mathbf{b}) + \log p(\mathbf{b}) + Cte$. Given the Gaussian laws $\mathbf{B} \sim \mathcal{N}(0, D_\nu)$ and $\mathbf{X}^{(0)} \sim \mathcal{N}(0, \sigma_0^2 I_n)$, this amounts to compute

$$\hat{\mathbf{b}} = \underset{\mathbf{b}}{\text{argmax}} -\frac{1}{\sigma_0^2} \|\mathbf{x} - U\mathbf{b}\|^2 - \mathbf{b}'D_\nu^{-1}\mathbf{b} , \quad (29)$$

which provides the expression $\hat{\mathbf{b}}$ in (23). Note that (29) is similar to the criterion of the Ridge regression [15].

5.2 Graph simulation

The simulation of a graph G consisting of m regulons : $G = \uplus_{k=1}^m R_k$, is done in three steps.

1. For each set of nodes V_k making up a regulon, a regulon graph $R_k = (V_k, E_k)$ is simulated.
2. At a larger scale, the m regulons are considered as m nodes of a graph, and thus an inter-regulon graph G^B is simulated.
3. The global graph G is obtained on the basis of these $m + 1$ graphs, as follows. For each regulon R_k , a regulator r_k is drawn uniformly at random in this regulon. This regulator regulates the regulon(s) $R_{k'}$ such that $k \sim k'$ in G^B ,⁵. The weight of the connections between r_k and nodes v in $R_{k'}$ are given by the probabilities of the Binomial law $\mathcal{B}(|R_{k'}|, p)$ where $0 < p < 1$. When a weight is below a threshold τ , for example 0.05, the weight is set to zero, then the probability distribution is renormalized. By ruling p and τ , one can modulate the number of edges between the regulator and the regulated unit. In this case, r_k regulates a subset of nodes in $R_{k'}$.

Acknowledgments

The referees are gratefully thanked. Their comments have improved the manuscript. The author is grateful to Alain Trouvé and Yong Yu for the experience we shared on the multi-scale decomposition of images, which has been an inspiration. The author thanks warmly Benno Schwikowski for valuable discussions about the adaptation of *Bacillus subtilis* to nutritional environments, and Xiaoyi Chen for her help in carrying out experiments on *Bacillus subtilis* data.

⁵A node $v \in V$ is regulated by another nodes v' if X_v is significantly correlated to $X_{v'}$.

References

- [1] Yong-Yeol Ahn, James P. Bagrow, and Sune Lehmann. Link communities reveal multiscale complexity in networks. *Nature*, 466(5):761–764, 2010.
- [2] Mikhail Belkin and Partha Niyogi. Semi-supervised learning on riemannian manifolds. *Machine Learning*, 56:209–239, 2004.
- [3] Andries E. Brouwer and Willem H. Haemers. *Spectra of graphs*. Springer, 2011.
- [4] Joerg Martin Buescher and *al.* Global network reorganization during dynamic adaptations of bacillus subtilis metabolism. *Sciences*, 335(6072):1099–1103, 2012.
- [5] Bernard Chalmond. *Modeling and Inverse Problems in Image Analysis*. Springer-Verlag, 2003.
- [6] Bernard Chalmond. Spatio-temporal graphical modeling with innovations based on multi-scale diffusion kernel. *Spatial Statistics*, 7:40–61, 2014.
- [7] Bernard Chalmond and Xiaoyi Chen. A graphical modeling to scan network activity at modular level. Technical report, Institut Pasteur /Cergy-Pontoise University, 2012.
- [8] Bernard Chalmond, Benjamin Francesconi, and Stephane Herbin. Using hidden scale for salient object detection. *IEEE Trans. on Image Processing*, 15(9):2644–2656, 2006.
- [9] Li Chen, Jianhua Xuan, Rebecca B. Riggins, Yue Wang, and Robert Clarke. Identifying protein interaction subnetworks by a bagging markov random field-based method. *Nucleic Acids Research*, 41(2), 2012.
- [10] Timoth e Cour, Florence B en ezite, and Jianbo Shi. Spectral segmentation with multiscale graph decomposition. In *CVPR*, 2005.
- [11] Guro Dorum, Lars Snipen, Margrete Solheim, and Solve Saebo. Smoothing gene expression data with network information improves consistency of regulated genes. *Statistical Applications in Genetics and Molecular Biology*, 10(1), 2011.
- [12] Marco A.R. Ferreira and Herbert K.H. Lee. *Multiscale Modeling : A Bayesian Perspective*. Springer, 2007.
- [13] Carlo Gaetan and Xavier Guyon. *Spatial Statistics and Modeling*. Springer-Verlag, 2009.
- [14] T.J. Hastie and R.J. Tibshirani. *Generalized Additive Models*. Chapman and Hall/CRC, 1999.
- [15] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The elements of statistical learning*. Springer, 2009.
- [16] Lasse Holmstrom, Leena Pasanen, Reinhard Furrer, and Stephan R. Sain. Scale space multiresolution analysis of random signals. *Computational Statistics and Data Analysis*, 55:2840–2855, 2011.
- [17] Trey Ideker, Owen Ozier, Benno Schwikowski, and Andrew F. Siegel. Discovering regulatory and signalling circuits in molecular interaction networks. *Bioinformatics*, 18(1):S233–S240, 2002.

- [18] Harri Kiiveri and Frank de Hoog. Fitting very large sparse gaussian graphical models. *Computational Statistics and Data Analysis*, 56:2626–2636, 2012.
- [19] Eric D. Kolaczyk. *Statistical Analysis of Network Data : Methods and Models*. Springer, 2009.
- [20] Risi Imre Kondor and John Lafferty. Diffusion kernels on graphs and other discrete input spaces. In Morgan Kaufmann, editor, *International Conference on Machine Learning*, pages 315–322, 2002.
- [21] Gert R. G. Lanckriet, Tijl De Bie, Nello Cristianini, Michael I. Jordan, and William Stafford Noble. A statistical framework for genomic data fusion. *Bioinformatics*, 20(16):2626–2635, 2004.
- [22] Peter Langfelder, Rui Luo, Michael C. Oldham, and Steve Horvath. Is my network module preserved and reproducible? *PLoS Computational Biology*, 7(1), 2011.
- [23] Ann B. Lee and Larry Wasserman. Spectral connectivity analysis. *Journal of the American Statistical Association*, 105, 2010.
- [24] Tony Lindeberg. Feature detection with automatic scale selection. *International Journal of Computer Vision*, 30(2):77–116, 1998.
- [25] Bojan Mohar. Some applications of laplace eigenvalues of graphs. In G. Hahn and G. Sabidussi, editors, *Graph Symmetry: Algebraic Methods and Applications*, volume Ser. C 497, pages 225, 275. Kluwer, 1997.
- [26] M. E. J. Newman. Finding community structure in networks using the eigenvectors of matrices. *Physical Review*, E(74), 2006.
- [27] Noa Novershtern, Aviv Regev, and Nir Friedman. Physical module networks: an integrative approach for reconstructing transcription regulation. *Bioinformatics*, 2011.
- [28] Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 22(8):888–905, 2000.
- [29] Ihor Smal, Marco Loog, Wiro Niessen, and Erik Meijering. Quantitative comparison of spot detection methods in fluorescence microscopy. *IEEE Trans. on Medical Imaging*, 29(2):282–301, 2010.
- [30] Daniel A. Spielman and Shang-Hua Teng. Spectral partitioning works: planar graphs and finite element meshes. In IEEE 1996, editor, *37th Symposium on Foundations of Computer Science*, pages 96–105, 1996.
- [31] Jian Sun, Maks Ovsjanikov, and Leonidas Guibas. A concise and provably informative multi-scale signature based on heat diffusion. In *Eurographics Symposium on Geometry Processing*, volume 28. Blackwell Publishing, 2009.
- [32] Liang Sun, Shuiwang Ji, , and Jieping Ye. Adaptive diffusion kernel learning from biological networks for protein function prediction. *BMC Bioinformatics* 9:162, 9(162), 2008.
- [33] Richard Szeliski. *Computer Vision: Algorithms and Applications*. Springer, 2010.
- [34] Kevin Thon, Havard Rue, Stein Olav Skovseth, and Fred Godtliebsen. Bayesian multiscale analysis of images modeled as gaussian markov random fields. *Computational Statistics and Data Analysis*, 56:49–61, 2012.

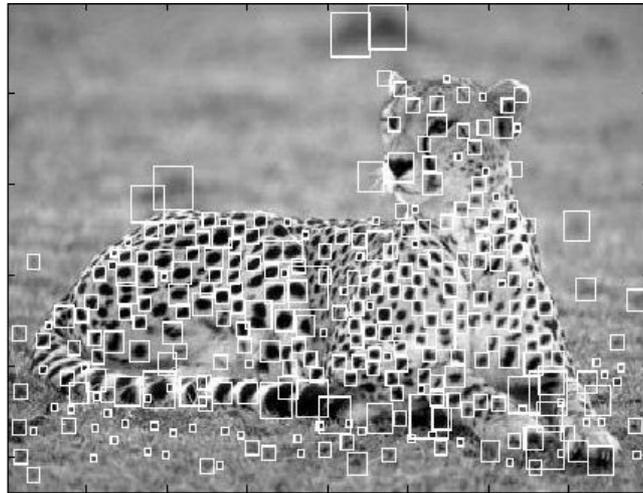


Figure 3: Scale-space blob detection on an image observed on a regular grid.

- [35] Igor Ulitsky and Ron Shamir. Identification of functional modules using network topology and high-throughput data. *BMC Systems Biology*, 1(8), 2007.
- [36] Ulrike von Luxburg. A tutorial on spectral clustering. *Statistics and Computing*, 17(4), 2007.
- [37] Grace Wahba. *Spline models for observational data*. SIAM, 1990.
- [38] Yuedong Wang and Grace Wahba. Bootstrap confidence intervals for smoothing splines and their comparison to bayesian confidence intervals. *J. Statist. Comput. Simulation*, 51:263–279, 1994.
- [39] Fan Zhang and Edwin R. Hancock. Graph spectral image smoothing using the heat kernel. *Pattern Recognition*, 41:3328 – 3342, 2008.

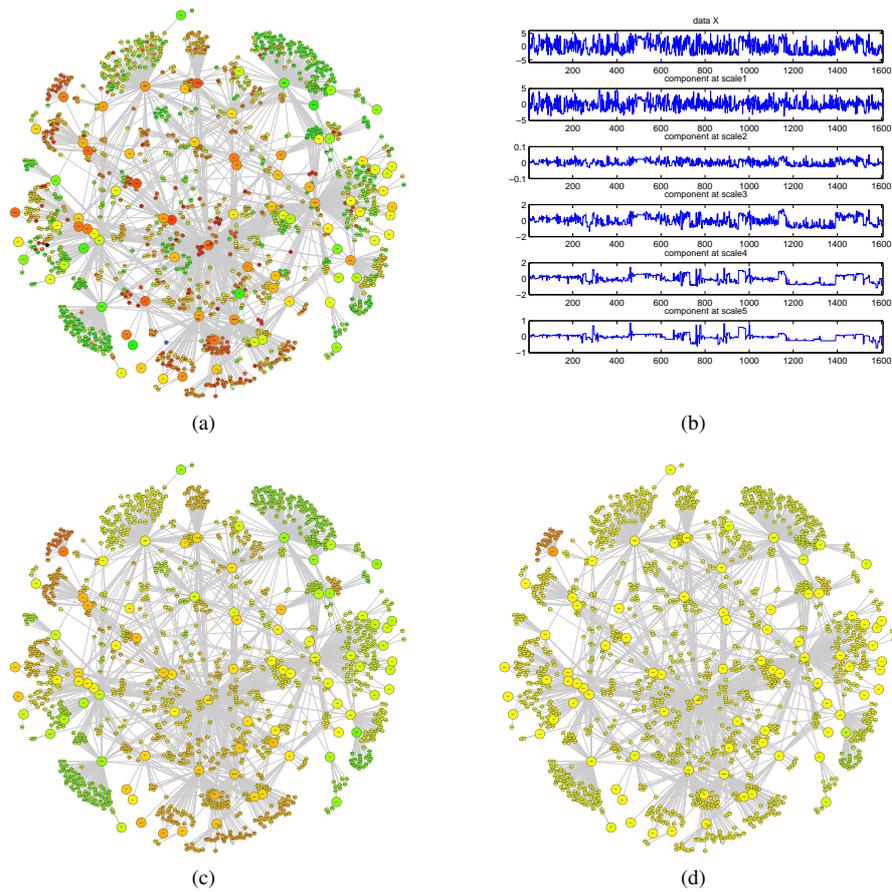


Figure 4: Multi-scale decomposition of gene expression network of *Bacillus subtilis* (from [7]). (a) Original data. (b) Multi-scale decomposition profiles (1-D display). (c-d) Scale components for $\lambda = 2$ and $\lambda = 16$, respectively. The decomposition has a structuring effect in terms of gene grouping. Despite the use of false colors, it is difficult to distinguish modules, unlike the case of an ordinary image as in Fig. 3.

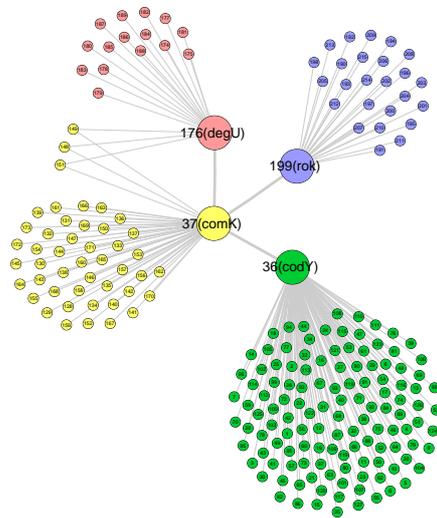


Figure 5: Graph partitioning of G : this graph shows four regulons and four regulators.

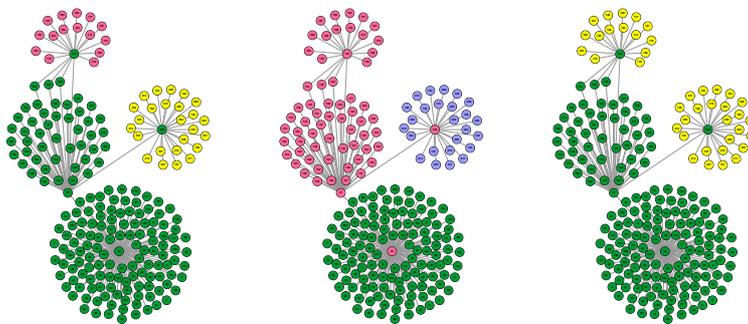


Figure 6: Module detection at three time points. This treatment was done without taking into account time dependence. Module detection yields respectively 3, 3 and 2 modules depending on $\mathbf{x}(t)$, $t = 1, 2, 3$.

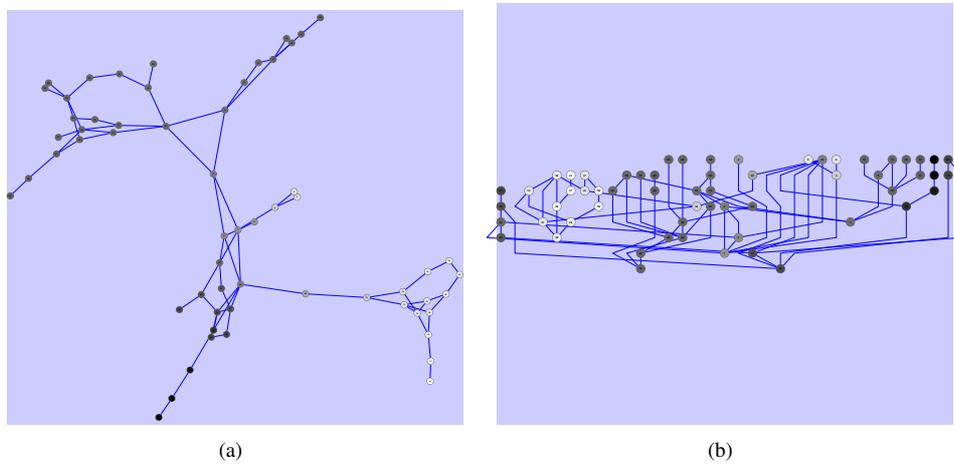


Figure 7: Same configuration $\{\mathbf{x}, G\}$ as in Fig. 8(c) but with two different displays : (a) "Edge weighted spring embedded" layout, (b) Hierarchical layout.

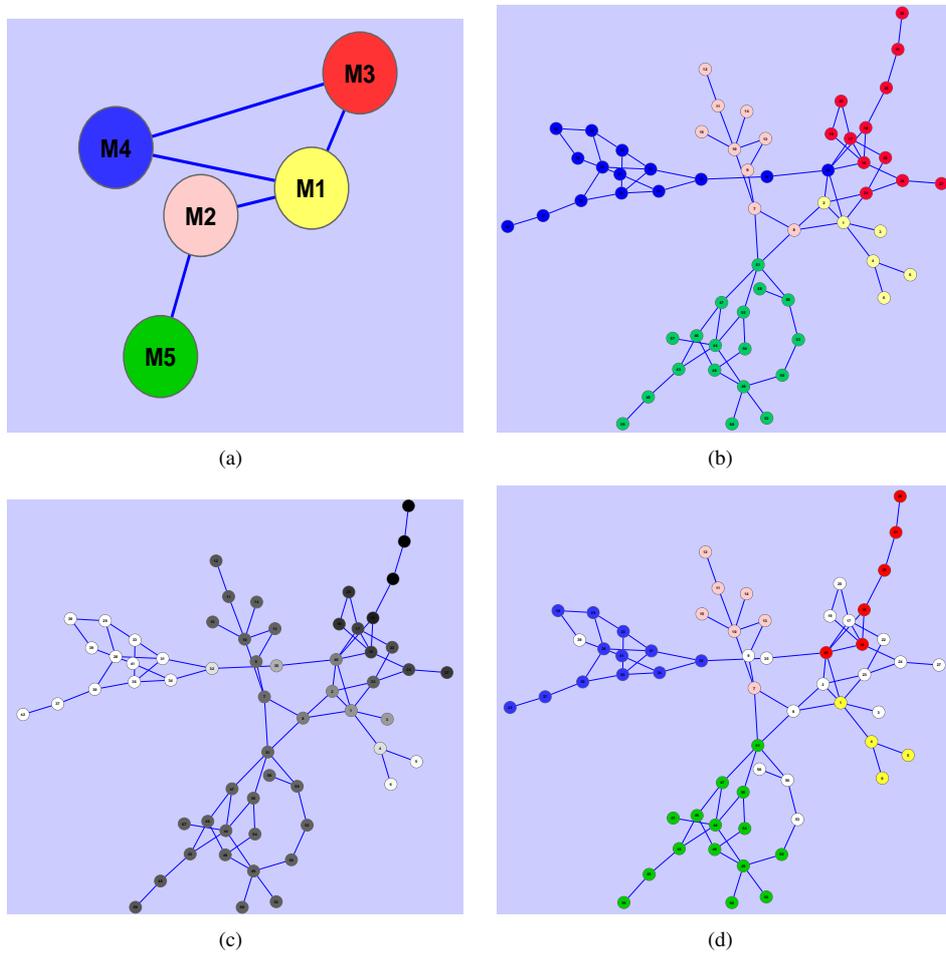


Figure 8: Experiment on simulated data. (a) The regulator graph G^B (between regulons). (b) The entire regulon graph G . (c) The observed random field \mathbf{x} (displayed with "Force directed" layout). (d) Module detection outcome: there is one detected module per regulon. Given the knowledge of the regulons, we can associate a regulon to each detected module.

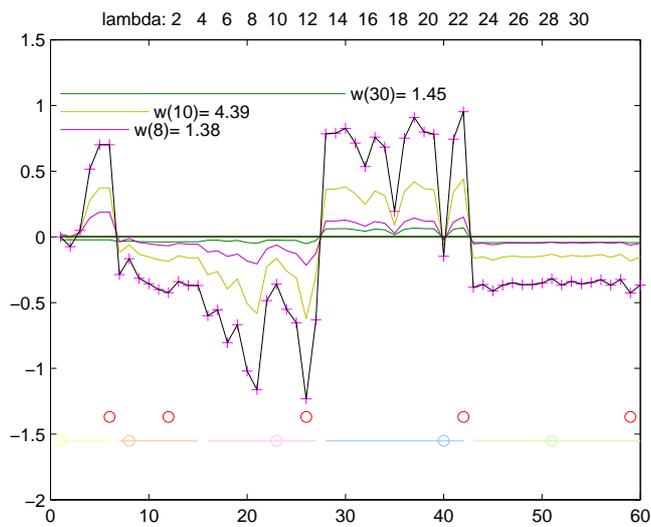


Figure 9: Module detection based on the multi-scale decomposition in Fig.10 restricted to Λ . The pink crosses show the 1-D profile of x . The color curves display the three main components for scales $\{8, 10, 30\}$. The red circles are the locations of the detected module centers. The color segments at the bottom of the figure locate regulons, their colors are identical to those in Fig.8(a-b-d).

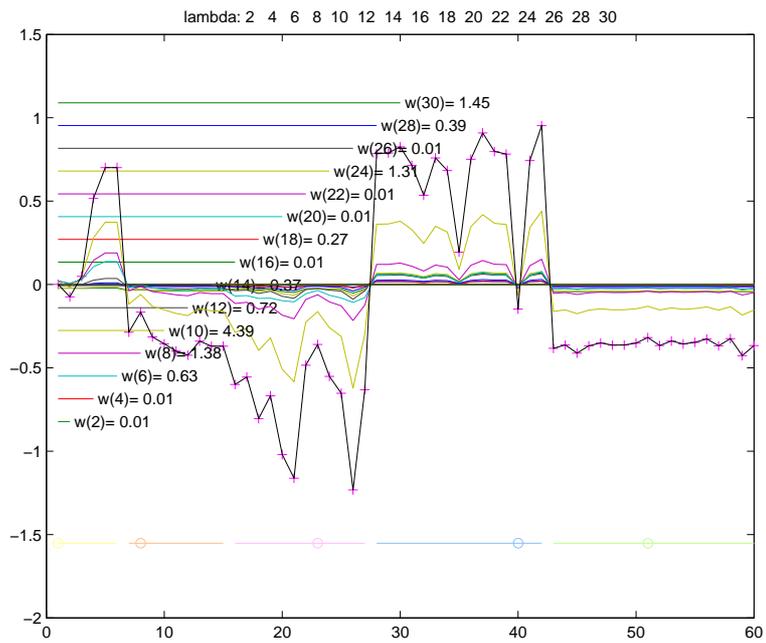


Figure 10: Statistical multi-scale decomposition for $\Lambda^0 = \{2, 4, \dots, 28, 30\}$. Scale selection with $\epsilon = 0.1$ selects $\Lambda = \{8, 10, 24, 30\}$. The black curve is the sum of all Λ^0 -components.

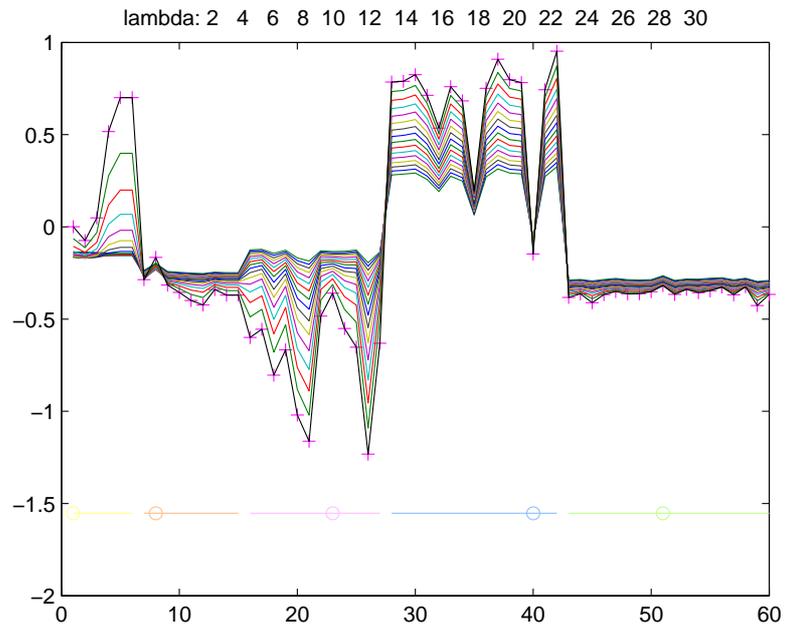


Figure 11: Ordinary scale-space decomposition.

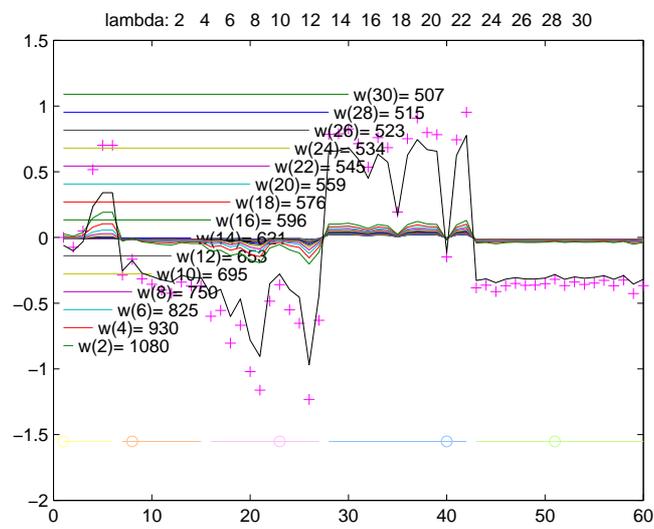


Figure 12: Adverse effect of an estimation made without $\log |\bar{K}_{\sigma,\Lambda}|$. All scales are nearly equal contributions and without informative value.