Computational Statistics and Data Analysis 120 (2018) 1–24

Contents lists available at ScienceDirect

Computational Statistics and Data Analysis

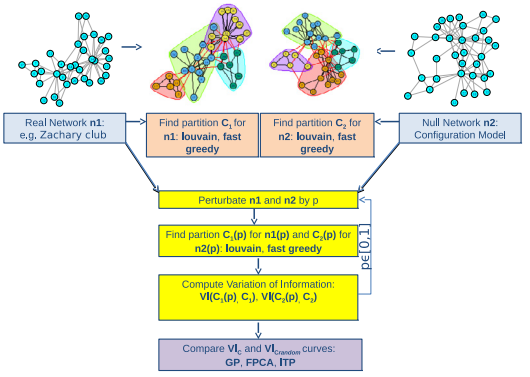
journal homepage: www.elsevier.com/locate/csda

Validation of community robustness

Annamaria Carissimo [a], Luisa Cutillo [b,c,*], Italia De Feis [d]

[a] Bioinformatics Core, TIGEM, Pozzuoli, Italy
[b] University "Parthenope" of Naples, Italy
[c] University of Sheffield, United Kingdom
[d] Istituto per le Applicazioni del Calcolo "Mauro Picone", Naples, Italy

GRAPHICAL ABSTRACT



ARTICLE INFO

Article history:
Received 21 March 2017
Received in revised form 25 October 2017
Accepted 27 October 2017
Available online 14 November 2017

Keywords:
Community
Network
Variation of information
Multiple testing

ABSTRACT

The large amount of work on community detection and its applications leaves unaddressed one important question: the statistical validation of the results. A methodology is presented that is able to clearly detect if the community structure found by some algorithms is statistically significant or is a result of chance, merely due to edge positions in the network. Given a community detection method and a network of interest, the proposal examines the stability of the partition recovered against random perturbations of the original graph structure. To address this issue, a perturbation strategy and a null model graph, which matches the original in some of its structural properties, but is otherwise a random graph, is specified. A set of procedures is built based on a special measure of clustering distance, namely Variation of Information, using tools set up for functional data analysis. The procedures determine whether the obtained clustering departs significantly from the null model. This strongly supports the robustness against perturbation of the algorithm

* Corresponding author at: University of Naples, Parthenope, Italy, and University of Sheffield, United Kingdom.
E-mail addresses: luisa.cutillo@parthenope.it, l.cutillo@sheffield.ac.uk (L. Cutillo).

https://doi.org/10.1016/j.csda.2017.10.006

used to identify the community structure. Results obtained with the proposed technique on simulated and real datasets are shown and discussed.

1. Introduction

Networks are mathematical representation of interactions among the components of a system and can be modelled by graphs. A graph $G = (V, E)$ consists of a collection of vertices V, corresponding to the individual units of the observed system, and a collection of edges E, indicating some relation between pairs of vertices.

Graphs modelling real systems, i.e. social, biological, and technological networks, display non trivial topological features. Indeed they present the properties that define a complex network: structural inhomogeneities, a broad degree distribution and distribution of edges locally inhomogeneous. In the study of complex networks, a network is said to have a community structure if the vertices can be divided in g groups, such that nodes belonging to the same group are densely connected and the number of edges between nodes of different groups is minimal.

The problem of community detection (graph partitioning) has been widely studied by researchers in a variety of fields, including statistics, physics, biology, social and computer science in the last 15 years. Finding communities within an arbitrary complex network can be a computationally difficult task. The number of communities, if any, within the network is typically unknown and the communities are often of unequal size and/or density. Despite these difficulties, however, several methods for community finding have been developed and employed with varying levels of success, see Coscia et al. (2011), Fortunato (2010), Goldenberg et al. (2010), Harenberg et al. (2014), Kolacyzk (2009) and Porter et al. (2009) for reviews.

Our work focuses on the problem of testing the robustness of the recovered partition of a given community detection method. In the following we provide a brief review of the state of the art of the literature addressing this problem. Although the remarkable work developed for community detection and its applications, the question of the significance of results still remains open. Our proposal represents a first attempt to statistically define the robustness of a clustering and hence cannot be directly compared to any of the following described methodologies.

2. State of the art

The modularity Q of Newman and Girvan (2004) was the first attempt to give an answer to this question. It is defined as the fraction of the edges that fall within the given groups minus the expected such fraction if edges were distributed at random and is based on the idea that a random graph is not expected to have a cluster structure. However, as pointed out in Fortunato (2010) and Karrer et al. (2008), there is an important limit. Precisely, networks with a strong community structure have high modularity, on the contrary high modularity does not imply networks with a community structure. Other authors, see Guimera et al. (2004) and Reichardt and Bornholdt (2007), suggested the use of a z-score to compare the maximum modularity of a graph to the maximum attainable modularity in purely random graphs of the same size and expected degree sequence. The problem is that the distribution of the null model, though peaked, is not Gaussian, causing false positives and false negatives.

A different approach was developed in Massen and Doye (2006), where the authors studied how canonical ensembles of network partitions depend on temperature to assess the significance and nature of the community structure obtained by algorithms that optimise the modularity. In this case $-Q$ plays the role of energy, i.e. at temperature T, the statistical weight of a given partition in the ensemble is proportional to $\exp(Q/T)$. Typically, as the temperature increases, there is a transition from low entropy/high Q partitions (significant cluster structure) to high entropy/low Q (random partitions). If there is strong community structure, the transition is sharp. The peak is broader for networks with weaker community structure, as there are more reasonable alternative partitions with intermediate values of Q, and so the transition occurs over a broader range of temperature. The authors also introduced an order parameter to measure the similarity of the sampled partitions at a given temperature, i.e. whether there is just a single partition with high Q or a number of competing partitions. Therefore, it is a useful tool to detect false positives. However the methodology is computationally onerous and cannot be easily generalised to other optimisation methods.

In Bianconi et al. (2009) the authors introduced the notion of entropy of graph ensembles to assess the relevance of additional information about the nodes of a network using the information that comes from the topology of the network itself. The indicator of clustering significance Θ introduced in the paper can also reveal statistical regularities that shed light on possible mechanisms underlying the network stability and formation.

In Lancichinetti et al. (2011) the authors presented the Order Statistics Local Optimisation Method (*OSLOM*), a technique based on the local optimisation of a fitness function, the C-score (Lancichinetti et al., 2010), expressing the statistical significance of a cluster with respect to random fluctuations. Given a subgraph C in a graph G, the C-score measures the probability that the number of links connecting a node to nodes in C, where C is embedded within a random graph, is higher than or equal to the value seen in the original graph G. This score permits to rank all the vertices external to C (in increasing order of the C-score), having at least one connection with C, and to calculate its order statistic distribution Ω. The minimum of Ω is the random variable whose cumulative is the score of the community C. To assess its significance a

threshold parameter P is fixed. The procedure is iterated to analyse the full network. The novelty of this approach is the local estimate of the significance, i.e. of single communities, not of partitions; on the contrary a serious limit is due to the lack of a data driven procedure to estimate P, indeed the authors fix its value to 0.1.

Recently, Wilson et al. (2014) proposed a testing based community detection procedure called Extraction of Statistically Significant Communities (ESSC). The ESSC procedure measures the statistical significance of connections between a single vertex and a set of vertices in undirected networks under a null distribution derived from the configuration model (Bender and Canfield, 1978). Given an observed network G_0 with n vertices and a vertex set B, the authors introduce the statistics $\hat{d}(u : B)$, measuring the number of edges between a vertex u and B in the random model $\hat{G}$, and show that $\hat{d}_n(u : B)$ is approximately binomial as $n \to \infty$ in the total variation distance between two probability mass functions. This permits to obtain the p-values of the null distribution using the binomial approximation and gives origin to an iterative deterministic procedure that recovers robust communities. The technique has some similarities with OSLOM, indeed both are extraction methods and use the configuration model as reference distribution, but differentiates because the probabilities have a closed form.

Another group of techniques was proposed in Gfeller et al. (2005), Karrer et al. (2008), Rosvall and Bergstrom (2010), and their conceiving was completely different from previous described methodologies. Indeed all the authors introduce a stochastic component in the network by perturbing the graph structure, measure the effect of the perturbation and compare it with the corresponding value for a null model graph. The basic idea is that a significant partition should not be altered by small modifications, as long as the modification is not too extensive. An interesting feature of these methods is their independence from the community detection technique adopted.

In this paper we present a methodology able to clearly detect if the community structure found by some algorithms is statistically significant or is a result of chance, merely due to edge positions in the network. Given a community detection method and a network of interest, our proposal examines the stability of the partition recovered against random perturbations of the original graph structure. To address this issue, following ideas from Karrer et al. (2008), we specify a perturbation strategy and a null model to build some procedures based on Variation of Information as stability measure. Given this measure we address the question of evaluating its significance. This permits to build the Variation of Information curve as a function of the perturbation percentage and to compare it with the corresponding null model curve using analysis tools set up for functional data analysis. Functional data analysis (FDA) is about the analysis of information on curves or functions and collects all the computational statistical methodologies set up for the analysis of data measured by some instruments on discrete grids, but representing curves. Moreover, what is unique about functional data is the possibility of also using information on the rates of change or derivatives of the curves Ramsay and Silverman (1997, 2002).

The rest of the paper is organised as follows. In Section 3 we introduce the proposed procedures based on Variation of Information and the functional data analysis techniques, including their detailed description. Section 4 shows the results achieved applying our methodology on simulated and real datasets. Conclusions and ideas for future research are drawn in Section 6.

3. Overall procedure

We propose to compare two different partitions on the same graph building on a special metric called Variation of Information (VI) (Meilă, 2007). We will show how to build a VI curve (VIc) comparing the partition of our original network and the partition of a perturbed version of the original network. We will describe a new hypothesis testing procedure to test if the VIc is significantly different from a random VI curve (referred to as VIc_{random}), obtained computing VI between the partition of a null random network and the partition of different perturbed version of such null network. The Variation of Information is an information theoretic criterion for comparing two partitions, or clusterings, of the same dataset introduced in Meilă (2007). It is a metric and measures the amount of information lost and gained in changing from clustering $\mathcal{C}$ to clustering $\mathcal{C}'$. The criterion makes no assumptions about how the clusterings were generated and applies to both soft and hard clusterings.

Given a dataset D of cardinality n and two clusterings $\mathcal{C}$ and $\mathcal{C}'$ of D, with K and K' non empty clusters, respectively, VI is defined as

$$VI\left(\mathcal{C}, \mathcal{C}'\right) = H\left(\mathcal{C}\right) + H\left(\mathcal{C}'\right) - 2I\left(\mathcal{C}, \mathcal{C}'\right), \tag{3.1}$$

where $H\left(\mathcal{C}\right)$ is the entropy associated with clustering $\mathcal{C}$

$$H\left(\mathcal{C}\right) = -\sum_{k=1}^{K} P(k) \log P(k), \tag{3.2}$$

and $I\left(\mathcal{C}, \mathcal{C}'\right)$ is the mutual information between $\mathcal{C}$ and $\mathcal{C}'$, i.e the information that one clustering has about the other

$$I\left(\mathcal{C}, \mathcal{C}'\right) = \sum_{k=1}^{K} \sum_{k'=1}^{K'} P\left(k, k'\right) \log \frac{P\left(k, k'\right)}{P\left(k\right) P\left(k'\right)}. \tag{3.3}$$

$P(k)$ is the probability of a point being in cluster C_k and $P\left(k, k'\right)$ is the probability that a point belongs to C_k in clustering $\mathcal{C}$ and to $C_{k'}$ in $\mathcal{C}'$, i.e. $P(k) = |C_k|/n$ and $P(k, k') = |C_k \cap C_{k'}|/n$.

Another equivalent expression for VI is

$$VI\left(\mathcal{C}, \mathcal{C}'\right) = H\left(\mathcal{C}|\mathcal{C}'\right) + H\left(\mathcal{C}'|\mathcal{C}\right). \tag{3.4}$$

The first term measures the amount of information about $\mathcal{C}$ that we loose, while the second measures the amount of information about $\mathcal{C}'$ that we gain, when going from clustering $\mathcal{C}$ to clustering $\mathcal{C}'$.

In the original paper Meilă (2007) the *VI* is juxtaposed to some indices and metrics for comparing clusterings, namely *Rand*, *adjustedRand*, *Fowlkes − Mallows*, *Jaccard* and *Wallace*, and the superiority of *VI* is discussed. More recently in Wade and Ghahrmani (2015) the authors introduced the use of *VI* as a loss function in the context of Bayesian cluster analysis showing several desirable properties.

VI metric is the basis of the hypothesis testing procedures we propose to establish the statistical significance of a recovered community structure in a complex network. Our original idea is to generate two different curves based on the VI measure and to statistically test their difference. The first curve *VIc* is obtained computing VI between the partition of our original network and the partition of different perturbed version of our original network. The second curve VIc_{random} is obtained computing VI between the partition of a null random network and the partition of different perturbed version of such null network. The comparison between the two VI curves turns the question about the significance of the retrieved community structure into the study of the stability/robustness of the recovered partition against perturbations. We expect that it must be robust to small perturbations, because if "small changes" in the network imply a completely different partition of the data, it means that the found communities are not trustworthy, and this cannot be due to the failure of the chosen algorithm for the community detection. Indeed the proposed testing procedure is independent from the clustering algorithm and it is easy to check if such a behaviour is due to it. To understand well this point we must consider the behaviour of the VI curve for networks having a real community structure and those having a very poor community structure. In the first case the VI curve starts at 0, when the perturbation level p is 0% (unperturbed graph), rises rapidly (perturbation level between 0% and 40%), then levels off when $50\% < p < 100\%$; in the second case the VI curve immediately grows up to a certain value and levels off that value. This last case means that whatever partition has been found, at each level of perturbation, the found community structure is a result of chance fluctuations and it is not plausible.

Obviously the set up of a testing procedure is more necessary for all cases where the community structure is moderate or weak and the behaviour of the VI curve could be similar to that of a random graph.

This explains why a small p-value is a good indicator that the community structure is reliable and can be considered believable, supporting greater evidence as the null model is able to reproduce closely local and global structural properties of real networks. Therefore, the choice of the null random network is really delicate, because we expect that it has the same structure of our original graph but with completely random edges.

This is why our choice relapses on the Configuration Model (Bender and Canfield, 1978) associated with the degree sequence of the observed graph $\mathbf{d} = \{d(1), \ldots, d(n)\}$ with vertex set $V = \{1, \ldots, n\}$, i.e. CM($\mathbf{d}$). It is a model able in capturing and preserving strongly heterogeneous degree distributions often encountered in real networks datasets and is the standard null model for empirical patterns. For a detailed discussion about random graphs and their use as null models we refer to Newman (2003) and Zweig (2016).

The CM($\mathbf{d}$) is a probability measure on the family of multi-graphs with vertex set V and degree sequence $\mathbf{d}$ that reflects, within the constraints of the degree sequence, a random assignment of edges between vertices. The generative form is simple: one can simply cut all the edges in the network, so every node still retains its degree by the number of half-edges or stubs emanating from it. The result will be an even number of half-edges. To create new networks with the same degree, one simply needs to randomly pair all the half-edges, creating the new edges in the network. The Configuration Model generates every possible graph with the given degree distribution with equal probability. Note that it naturally creates networks with multiple edges between nodes and self-connections between nodes. If such networks are unacceptable, one can reject those samples and try the algorithm again, repeating until one obtains a network without multiple or self-connections. In order to emphasise the importance of using a null random model that corresponds closely to the original network, we also explored the *dk* null random model provided in Orsini and others (2015). In the *dk* model a complete set of basic characteristics of the network structure, namely a *dk − series*, is employed to generate *dk*-random graphs whose degree distributions, degree correlations and clustering are as in the corresponding real network. To this end in our simulation study, we used the implementation *RandNetGen* of the *dk* model available on *github* (https://github.com/polcolomer/RandNetGen), along with the CM, as an alternative null model (option $-dk$ 2.5). Discussion of the results is addressed in Section 4. As for the perturbation strategy adopted, this will be described in Section 3.4. The basic steps of our method can be summarised in Algorithm 1.

Note that the variation of p from 0 to 1 induces an intrinsic order to the data structure as in temporal data and can be treated as time point. Moreover, as it will be described in Section 3.4, we generate many perturbed graphs (i.e. 10) for each different level of p and these are considered as replicates per time points in our strategy. The permutation step indeed requires the core computational time that decreases with the sparseness of the network under study. Of course for every permutation level p, after the permutation step, the selected clustering method is applied both on the perturbed original network and on the perturbed configuration model. Hence the choice of the community extraction method may affect the computation time and also the scalability of the overall approach. In this paper we will use two literature algorithms, namely

Data: a given network N, with vertex set $V = \{1, \ldots, n\}$ and degree sequence $\mathbf{d} = \{d(1), \ldots, d(n)\}$, a chosen method M, a set of perturbation levels $p \in [p_{min}, \ldots, p_{max}] \subseteq [0, 1]$, a null random model, i.e. CM($\mathbf{d}$).
Result: Build VIc and VIc_{random} curves as functions of perturbation level p and statistically test for *"the difference"* between them.
Initialisation: find a partition $\mathcal{C}$ of N and $\mathcal{C}_{rand}$ of CM($\mathbf{d}$) by the chosen method M;
while $p \in [p_{min}, \ldots, p_{max}]$ **do**
- perturb both N and CM($\mathbf{d}$) edges by the same percentage p, preserving the original graphs degree distributions;
- find a partition $\mathcal{C}'$ and $\mathcal{C}'_{rand}$ for the perturbed networks by the same method M;
- compute the $VIc(p) = VI(\mathcal{C}, \mathcal{C}')$;
- compute $VIc_{random}(p) = VI(\mathcal{C}_{rand}, \mathcal{C}'_{rand})$;
- $p++$

end
Testing: statistically test *"the difference"* between the VIc and VIc_{random} curves.

Algorithm 1: Overall Procedure

Fast Greedy (Clauset et al., 2004) and *Louvain* (Blondel et al., 2008), that will be described in section 4. The CPU time for a 2000 nodes sparse network on a Unix node with 2 Intel Xeon X5675 3.07 GHz processor and 48 GB DDR3 1333 MHz of ram, was 48 min when using *Louvain* and 58 min when using *Fast Greedy* as clustering methods.

The testing step of the above procedure is achieved by a functional data analysis approach aiming to test if the two groups of curves represent "the same process" or "different processes". The testing procedure we rely on is based on a tool set up for time course microarray, namely Gaussian Process (GP) regression (Kalaitzis and Lawrence, 2011). Aim of the GP regression in the context of gene expression data is to identify differentially expressed genes in a one-sample time course microarray experiment, i.e. to detect if the profile has a significant underlying signal or the observations are just random fluctuations. In this case we reformulate the testing problem working on $\log_2(VIc/VIc_{random})$, as described in Section 3.1.

In order to show that our approach is robust with respect to the testing procedure, we also display the overall results achieved when using other two approaches as described respectively in Sections 3.2 and 3.3. Indeed, we can look at the two measured VI curves as independent realisations of two underlying processes say X_1 and X_2 observed with noise on a finite grid of points $p \in [0, 1]$ and to test the null hypothesis

$$H_0 : \mathrm{X}_1 \overset{\mathrm{d}}{=} \mathrm{X}_2, \tag{3.5}$$

versus the alternative hypothesis

$$H_1 : \mathrm{X}_1 \overset{\mathrm{d}}{\neq} \mathrm{X}_2,$$

where $\overset{d}{=}$ means that the processes on either side have the same distribution.

Then, as described in Section 3.2, taking advantage of the Karhunen–Loève expansion we explore the methodology developed in Pomann et al. (2016) based on Functional Principal Components Analysis (FPCA) to test (3.5).

On the contrary, the approach described in Section 3.3 addresses a domain-selective inferential procedure, providing an interval-wise non parametric functional testing (Pini and Vantini, 2016), able not only to assess (3.5), but also to point out specific differences.

We will briefly describe GP regression, FPCA and interval-wise functional testing in the following sections. We would like to point out that our overall procedure provides a workflow to validate a community structure under different perspectives that can be investigated in dependence of the specific real problem dealt with. The description of the three different testing procedure is functional to the understanding of our overall procedure and in particular how we exploit the theory underling each single methodology to compare the curves VIc and VIc_{random}. Hence we will summarise the three testing procedures to highlight the key connection to our testing problem. We choose to provide a review of each single methodology to provide awareness of the differences between the three procedures and their link to our testing problem. Even if addressing the same problem the techniques are not equivalent. We refer to the original papers for any theoretical property of such testing procedures, including type I/II error study. We want to stress that the original contribution of our proposal is summarised in the algorithm pack depicted in the above frame.

3.1. GP regression

In this section we briefly summarise the methodology proposed in Kalaitzis and Lawrence (2011), where the authors present an approach to estimate the continuous trajectory of gene expression time-series from microarray through GP regression.

Briefly we recall that a Gaussian process is the natural generalisation of a multivariate Gaussian distribution to a Gaussian distribution over a specific family of functions. More precisely, as defined in Rasmussen and Williams (2006), a Gaussian process is a *collection of random variables, any finite number of which have a joint Gaussian distribution* and is completely

specified by its mean function and its covariance function. If we define the mean function $m(x)$ and the covariance function $k(x, x')$ of a real process $f(x)$ as:

$$m(x) = E[f(x)],$$
$$k(x, x') = E[(f(x) - m(x))(f(x') - m(x'))],$$

then we can write the GP as

$$f(x) \sim \mathcal{GP}(m(x), k(x, x')). \tag{3.6}$$

The random variables $\mathbf{f} = (f(X_1), \ldots, f(X_n))^T$ represent the value of the function $f(x)$ at time locations $(X_i)_{i=1,\ldots,n}$, being $f(x)$ the true trajectory/profile of the gene. Assuming $f(x) = \Phi(x)^T\mathbf{w}$, where $\Phi(x)$ are projection basis functions, with prior $\mathbf{w} \sim N(\mathbf{0}, \sigma_{\mathbf{w}}^2\mathbf{I})$, we have

$$E[f(x)] = \Phi(x)^T E[\mathbf{w}] = 0, \tag{3.7}$$
$$E[f(x)f(x)'] = \sigma_{\mathbf{w}}^2\Phi(x)^T\Phi(x), \tag{3.8}$$
$$f(x) \sim \mathcal{GP}(0, \sigma_{\mathbf{w}}^2\Phi(x)^T\Phi(x)). \tag{3.9}$$

Since observations are noisy, i.e. $\mathbf{y} = \mathbf{\Phi w} + \boldsymbol{\varepsilon}$, with $\mathbf{\Phi} = (\Phi(X_1)^T, \ldots, \Phi(X_n)^T)$, assuming that the noise $\boldsymbol{\varepsilon} \sim N(\mathbf{0}, \sigma_n^2\mathbf{I})$ and using Eqs. (3.7)–(3.8), the marginal likelihood

$$p(\mathbf{y}|\mathbf{x}) = \int p(\mathbf{y}|\mathbf{x}, \mathbf{w})p(\mathbf{w})d\mathbf{w},$$

becomes

$$p(\mathbf{y}|\mathbf{x}) = \frac{1}{(2\pi)^{n/2}\left|\mathbf{K_y}\right|^{1/2}}\exp\left(-\frac{1}{2}\mathbf{y}^t{\mathbf{K_y}}^{-1}\mathbf{y}\right), \tag{3.10}$$

with $\mathbf{K_y} = \sigma_{\mathbf{w}}^2\mathbf{\Phi\Phi}^T + \sigma_n^2\mathbf{I}$.

In this framework the hypothesis testing problem can be reformulated, over the perturbation interval [0, 1], as:

$$H_0 : \log_2\frac{VIc(x)}{VIc_{random}(x)} \sim \mathcal{GP}(0, k(x, x')),$$

against

$$H_1 : \log_2\frac{VIc(x)}{VIc_{random}(x)} \sim \mathcal{GP}(m(x), k(x, x')).$$

The marginal likelihood derived from Eq. (3.10), enables then to compare or rank different models by calculating the Bayes Factor (BF). More specifically the BF is approximated with a log-ratio of marginal likelihoods of two GPs, each one representing the hypothesis of differential (the profile has a significant underlying signal) and non differential expression (there is no underlying signal in the profile, just random noise). The significance of the profiles is then assessed based on the BF.

3.2. Functional principal component testing

In this section we briefly summarise the approach proposed in Pomann et al. (2016) to test the hypothesis (3.5) when the observed data are realisations of the curves at finite grids and possibly corrupted by noise. Their motivating application is a diffusion tensor imaging study, where the objective is to compare white matter track profiles between healthy individuals and multiple sclerosis patients. The authors introduce a novel framework based on functional principal component analysis (*FPCA*) of an appropriate mixture process, referred to as marginal *FPCA*. The statistical framework for this problem assumes to observe data arising from two groups, namely $\{(t_{1ij}, Y_{1ij}) : j = 1 \ldots m_{1i}\}_{i=1}^{n_1}$ and $\{(t_{2ij}, Y_{2ij}) : j = 1 \ldots m_{2i}\}_{i=1}^{n_2}$, where $t_{1ij}, t_{2ij} \in T$, a compact interval that in our case is $T = [0, 1]$ (time plays the role of perturbation level p). It is assumed that the Y_{1ij} and the Y_{2ij} are independent realisations of two underlying processes observed with noise on a finite grid of points:

$$Y_{1ij} = X_{1ij} + \epsilon_{1ij},$$
$$Y_{2ij} = X_{2ij} + \epsilon_{2ij},$$

where $X_{1ij} \sim^{IID} X_1(\cdot)$ and $X_{2ij} \sim^{IID} X_2(\cdot)$ are independent and square integrable random functions over T, for some underlying (latent) random processes X_1 and X_2. It is assumed that X_1 and X_2 are second-order stochastic processes with mean functions assumed to be continuous and covariance functions assumed to be continuous and positive semidefinite, both being unknown. The measurement errors $\{\epsilon_{1ij}\}$ and $\{\epsilon_{2ij}\}$ are independent and identically distributed (*IID*), with zero mean and variances σ_1^2 and σ_2^2, respectively. The authors exploit the truncated Karhunen–Loève expansion of the mixture process $X(\cdot)$ of $X_1(\cdot)$ and $X_2(\cdot)$ with mixture probabilities p and $1 - p$. Let Z a binary random variable taking values in $\{1, 2\}$ with

$P(Z = 1) = p$, then $X_1(\cdot) = E[(\cdot)/Z = 1]$ and $X_2(\cdot) = E[(\cdot)/Z = 2]$. Let us consider the truncated Karhunen–Loève expansion of $X(\cdot)$ and define $X_Z^K(t) = \mu(t) + \sum_{k=1}^K \xi_{Zk}\Phi_k(t), Z = 1, 2$, testing hypothesis (3.5) reduce to testing if the *FPC* scores $\{\xi_1^k\}_{k=1}^K$ and $\{\xi_2^k\}_{k=1}^K$ have the same distribution:

$$H_0^K : \{\xi_1^k\}_{k=1}^K \stackrel{d}{=} \{\xi_2^k\}_{k=1}^K. \tag{3.11}$$

In practice the authors consider K null hypothesis given the finite truncation level and propose a multiple two-sample univariate test, the *Anderson-Darling* (*AD*) statistic (Petit, 1976), combined with a multiple-comparison adjustment. The authors propose a Bonferroni correction, a procedure which controls the probability of erroneously rejecting even one of the true null hypotheses, the Family Wise Error Rate (FWER). In this case hypothesis (3.11) is rejected if

$$\min_{1\le k\le K} p_k \le \alpha/K, \tag{3.12}$$

where p_k is the p-value that is obtained by using the chosen univariate two-sample test for each H_0^k.

The false discovery rate (FDR), suggested in Benjamini and Hochberg (1995) is a different point of view for how the errors in multiple testing could be considered. The FDR is the expected proportion of erroneous rejections among all rejections. If all tested hypotheses are true, controlling the FDR controls the traditional FWER. But when many of the tested hypotheses are rejected, indicating that many hypotheses are not true, the error from a single erroneous rejection is not always as crucial for drawing conclusions from the family tested, and the proportion of errors is controlled instead. Using the individual testing statistics proposed in Pomann et al. (2016) we will therefore adopt this FDR approach to adjust our tests for multiplicity.

Note that this procedure is designed for a more general framework in which the two curves *VI* and *VIc* can be observed at different time points (i.e. $p \in [0, 1]$).

3.3. Interval-wise functional testing

In the following we will briefly review the Interval-wise Functional testing procedure (*ITP*) proposed by Pini and Vantini (2016), where the authors develop a non-parametric domain-selective inferential methodology for functional data embedded in the $L^2(T)$ space (where T is any limited open interval of $\mathbb{R}$) to test (3.5). Their technique is not only able to assess the equality in distribution between functional populations, but also to point out specific differences. Their procedure is based on the following three steps:

1. Basis Expansion: functional data are projected on a functional basis (i.e. Fourier or B-splines expansion);
2. Interval-Wise Testing: statistical tests are performed on each interval of basis coefficients;
3. Multiple Correction: for each component of the basis expansion, an adjusted p-value is computed from the p-values of the tests performed in the previous step.

More in detail, let us assume to observe two independent samples of sizes n_1 and n_2 of independent random functions on a separable Hilbert space $y_{ij}(t), i = 1, \dots, n_j, j = 1, 2$.

In the first step, data are projected on a finite-dimension subspace generated by a reduced basis $y_{ij}(t) = \sum_{k=1}^p c_{ij}\Phi^{(k)}(t)$, where integer p represents the dimension. It follows that each of the $n = n_1 + n_2$ units can be represented by means of the corresponding p coefficients $\{c_{ij}^{(k)}\}$, $k = 1, \dots, p$; moreover, for each k, $c_{i1}^{(k)}$, $i = 1, \dots, n_1$ and $c_{i2}^{(k)}$, $i = 1, \dots, n_2$ are independent, and $c_{i1}^{(k)} \sim C_1^{(k)}$ and $c_{i2}^{(k)} \sim C_2^{(k)}$ where $C_1^{(k)}$ and $C_2^{(k)}$ denote the (unknown) distributions of the kth basis coefficient in the two populations.

In the second step, the authors build a family of multivariate tests for

$$H_0^{(\mathbf{k})} = \cap_{k\in\mathbf{k}} H_0^{(k)}, \quad H_0^{(k)} : C_1^{(k)} \stackrel{d}{=} C_2^{(k)},$$

$k = 1, \dots, p$ and $\mathbf{k}$ is a vector of successive indexes in $\{1, \dots, p\}$. In addition the authors add the multivariate tests on the complementary sets of each interval, i.e., do also test each hypothesis $H_0^{(\mathbf{k^c})} = \cap_{k\notin\mathbf{k}} H_0^{(k)}$. The tests are performed by the Nonparametric Combination Procedure (NPC), see Pesarin and Salmaso (2010), that constructs multivariate permutation tests by means of combining univariate-synchronised permutation tests.

In the third step the authors obtain the adjusted p-value for the kth component $\lambda_{\text{ITP}}^{(k)}$ by computing the maximum over all p-values of interval-wise tests whose null hypothesis implies $H_0^{(k)}$:

$$\lambda_{\text{ITP}}^{(k)} = \max\left(\max_{\mathbf{k}\ s.t.\ k\in\mathbf{k}} \lambda^{(\mathbf{k})}, \max_{\mathbf{k^c}\ s.t.\ k\in\mathbf{k^c}} \lambda^{(\mathbf{k^c})}\right),$$

and prove that, if we reject the kth adjusted p-value $\lambda_{\text{ITP}}^{(k)} \le \alpha$, then, for any interval $\mathbf{k}$ s.t. $H_0^{(k)}$ is true $\forall k \in \mathbf{k}$, the probability of rejecting any $H_0^{(k)}$ is lower or equal to α. This property reads interval-wise control of the FWER.

3.4. Perturbation strategy

Mimicking the approach proposed by Karrer et al. (2008) and Cutillo et al. (2012), we restrict our perturbed networks to having the same numbers of vertices and edges as the original unperturbed network, hence only the positions of the edges change. In other words we apply a Degree Preserving Randomisation. Our perturbation strategy relies on the *rewire* function belonging to the *R* package *igraph*, using the option *keeping_degseq*. Moreover, we expect that a network perturbed by only a small amount has just a few edges moved in different communities, while a maximally perturbed network produces completely random clusters.

In Karrer et al. (2008) the perturbation strategy is achieved by removing each edge with a certain probability α and replacing it with another edge between a pair of vertex (i, j) chosen at random with a probability proportional to the degree of i and j. This perturbation scheme generates networks that have the same number of edges as the original and in which the expected degrees of vertices are the same as the original degrees.

Our perturbation strategy consists in randomly rewiring a percentage p of edges while preserving the original graph's degree distribution. The rewiring algorithm indeed chooses two arbitrary edges in each step (e.g. (a, b) and (c, d)) and substitutes them with (a, d) and (c, b), if they do not already exists in the graph. The algorithm does not create multiple edges.

A null percentage of permutation $p = 0$ corresponds to the original unperturbed graph, while $p = 1$ corresponds to the maximal perturbation level. Varying the percentage p from 0 (original graph) to 1 (maximal perturbation), many perturbed graph are generated and compared to the partition on the original graph by means of *VI*. Indeed we generated 10 perturbed graph for each different level of $p \in [0, 1]$. Then, from each of the obtained graphs, we generated other 10 graphs rewiring 1% of edges each time. Hence resulting in 100 graphs for each level of $p \in [0, 1]$. In our setting we chose 20 levels of p.

Since the degree distribution is generally inferred directly from the observed graph, i.e. from the data, which are only a part of some hypothetical "true" random graph that is never fully observed, perhaps it could be interesting to extend further our strategy. Indeed, we could perturb the original graph choosing a broader class of networks whose degree distribution is within the bootstrapped confidence interval of the original degree distribution (Gel et al., 2017). As highlighted by the authors of the paper, there are a lot of methods to estimate the degree distribution directly from a graph, i.e. from the data, but what is missing is quantification of estimation uncertainty. So they develop the Fast Patchwork Bootstrap (FPB) algorithm to estimate network degree distribution and quantify a confidence interval under the assumption that the network distribution is involution invariant, i.e. selecting at random any vertex, the rest of the connected network is probabilistically the same. The authors use the "blocking" technique developed for resampling of space and time dependent processes and adapt it to networks: first select randomly vertices, then build local vicinities or patches and then resample vertices within local patches.

Another solution could be to use the technique proposed in De Vico Fallani et al. (2014), where the authors propose a method to replicate structural features of complex networks based on non parametric bootstrapping to improve the performance of spectral community detection algorithms. They consider a non parametric resampling of the transition matrix associated with an unbiased random walk on the graph. In particular the method builds different replicates of the transition matrix of the network, estimates an average distance matrix, whose elements correspond to the expected spectral distances between pairs of nodes of the graph, averaged over the ensemble of replicates and uses a standard hierarchical clustering algorithm on the obtained distance matrix. The method uses the idea that the aggregation of information about different replicates should allow to obtain more accurate and robust partitions than the one found from the original observed network.

We stress again that all the proposed strategies arose from the awareness that a real-world network is just a single observation drawn from an unknown distribution of graphs having certain characteristics. As a consequence, there is no predefined way to assess the statistical variability of any network property, including the presence and composition of communities except to consider random network ensembles, i.e., sets of graphs obtained from the original network by keeping fixed some structural properties.

4. Results

The overall procedure proposed in the present paper was implemented in *R* and validated both on simulated and real networks as will be described in the following Sections 4.1 and 4.3. In Fig. 1 we depict a summary of the proposed overall procedure main steps. For each of the analysed networks (either simulated or real) we performed the community extraction step, using some tools embedded in the *R* package *igraph*. We chose *igraph* because it provides an implementation of graph algorithms able to fast identifying community structures in large graphs. In particular we used two community extraction functions, one based on a greedy optimisation of the modularity (*cluster_fast_greedy*) and another based on a multi-level optimisation of the modularity (*cluster_louvain*). More specifically *cluster_fast_greedy* implements the hierarchical agglomeration algorithm for detecting community structure described in Clauset et al. (2004) and *cluster_louvain* is based on the hierarchical approach proposed in Blondel et al. (2008). Both these methods enables for an automatic definition of the optimal number of communities, are specific for large networks and are based on the optimisation of the modularity. The techniques are briefly summarised in the following.

As regards the testing methodologies we used the bioconductor package gprege available at https://www.bioconductor.org/packages/release/bioc/html/gprege.html for the GP regression, the R code from the professor Staicu's web-site http://www4.stat.ncsu.edu/~staicu/ for the Functional Principal Component test and the R package fdatest available at https://cran.r-project.org/web/packages/fdatest/index.html for the Interval-wise Functional test.

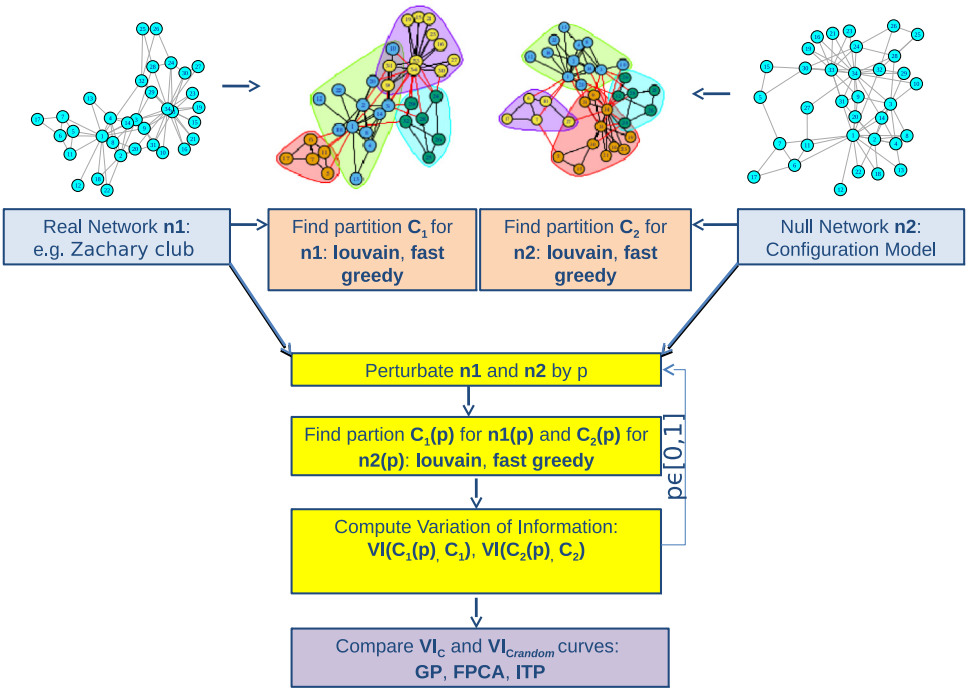


Fig. 1. Overall procedure map.

Fast greedy method

Fast Greedy is the modularity optimisation algorithm introduced by Clauset et al. (2004). This method is essentially a fast implementation of a previous technique proposed by Newman (Newman and Girvan, 2004). Starting from a set of isolated nodes, the links of the original graph are iteratively added such to produce the largest possible increase of the modularity. Adding a first edge to the set of disconnected vertices reduces the number of groups forming a new partition of the graph. The edge is chosen such that this partition gives the maximum increase (minimum decrease) of modularity with respect to the previous configuration. All other edges are added based on the same principle. At each iteration step, the variation of modularity given by the merger of any two communities of the running partition is computed and the best merger chosen. The fast version of Clauset, Newman and Moore, which uses more efficient data structures, has a complexity of $O(N\log_2 N)$ on sparse graphs.

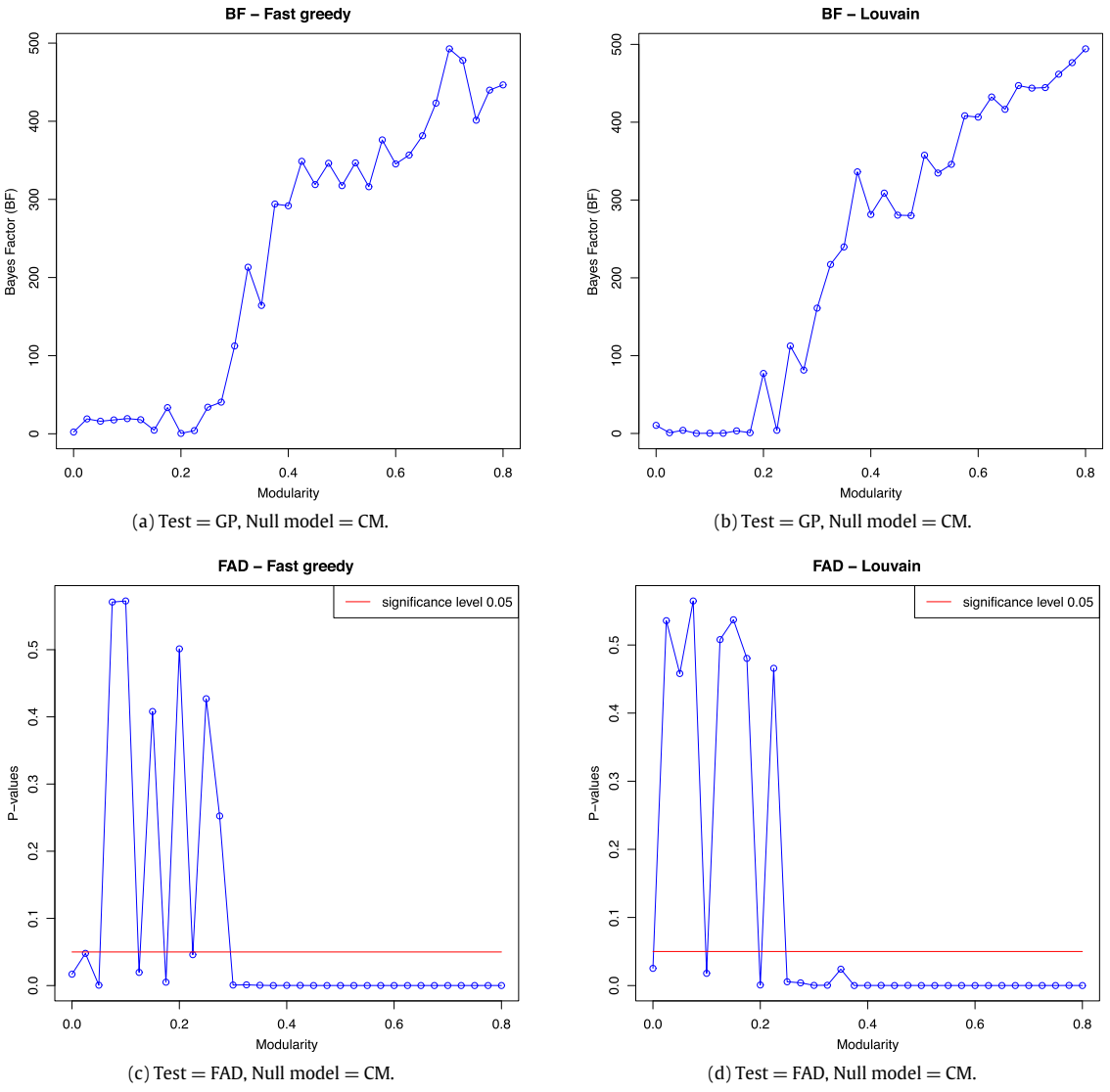
Louvain method

Louvain method is the fast modularity optimisation by Blondel et al. (2008). This technique consists of two steps, executed alternatively. Initially, each node is in its own community. In step 1, nodes are considered one by one, and each one is placed in the neighbouring community (including its own) that maximises the modularity gain. This is repeated until no node is moved (the obtained decomposition provides therefore a local optimisation of Newman–Girvan modularity). After a partition is identified in this way, in step 2 communities are replaced by super-nodes, yielding a smaller weighted network where two super-nodes are connected if there is at least an edge between vertices of the corresponding communities. The two steps of the algorithm are then repeated until modularity (which is always computed with respect to the original graph) does not increase any further.

As pointed out in Fortunato (2010), this method offers a fair compromise between the accuracy of the estimate of the modularity maximum, which is better than that delivered by greedy techniques like the one by Clauset et al. above, and computational complexity, which is essentially linear in the number of links of the graph.

4.1. Application to simulated data

In order to show the ability of our method to validate a network clustering, we applied it to modular random network graphs generated using the model implemented in Sah et al. (2014). The model generates undirected, simple, connected graphs with prescribed degree sequences and a specified level of community structure, while maintaining a graph structure that is otherwise as random (uncorrelated) as possible over a broad range of distributions of network degree and community



(a) Test = GP, Null model = CM.

(b) Test = GP, Null model = CM.

(c) Test = FAD, Null model = CM.

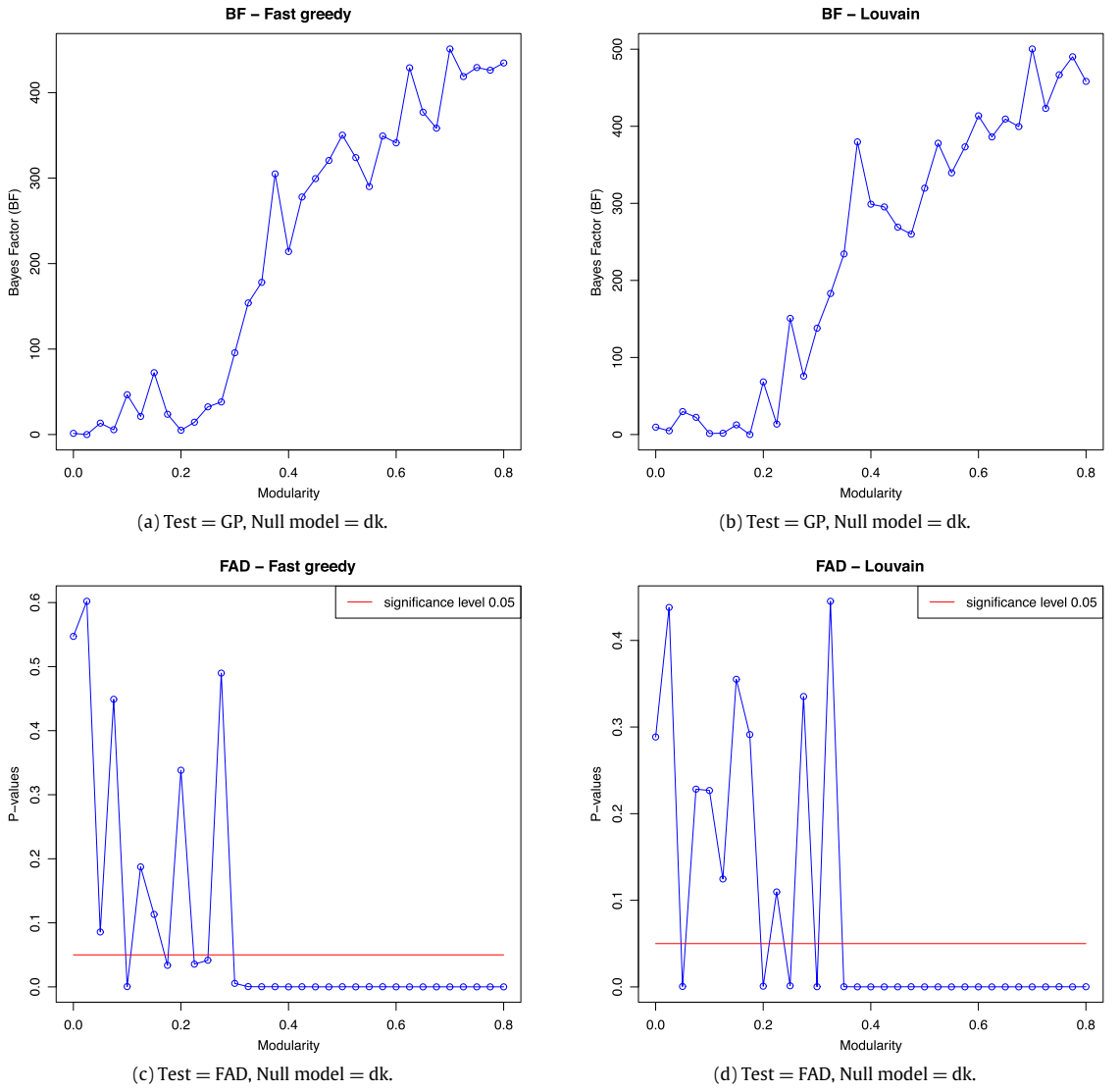
(d) Test = FAD, Null model = CM.

Fig. 2. Summary GP and FAD results on a grid of 33 modularity values in [0, . . . , 0.8], grid step 0.025, using CM as null model. In the first row the BFs from GP testing procedure are reported when using Fast Greedy (a) and Louvain (b) as community extraction methods. In the second row the *p*-values from the FAD testing procedure are reported when using Fast Greedy (c) and Louvain (d) as community extraction methods. The red line corresponds to 0.05 *p*-value. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

size. The model in Sah et al. (2014) is specified by the network size, the average network degree, the number of modules, the modularity, the degree distribution and the module size distribution. The authors propose an algorithm based on the following four steps:

1. Assign arbitrarily the nodes to modules whose sizes are sampled from the specified module size distribution;
2. Assign degrees to each node sampling a degree sequence from the specified degree distribution. The within-degrees are assigned to each nodes assuming that the within-degree distribution follows the class of the specified degree distribution;
3. Connect between-edges using a modified version of the Havel–Hakimi algorithm (Hakimi, 1962; Havel, 1955). The connections are then randomised by rewiring through double-edge swaps (Gkantsidis et al., 2003) ;
4. Connect within-edges using the standard Havel–Hakimi algorithm (Hakimi, 1962; Havel, 1955). The connections are then randomised by rewiring through double-edge swaps (Gkantsidis et al., 2003).

The generated graph results also to be as random as possible, to contain no self loops (edges connecting a node to itself), multi-edges (multiple edges between a pair of nodes), isolate nodes (nodes with no edges), or disconnected components



(a) Test = GP, Null model = dk.

(b) Test = GP, Null model = dk.

(c) Test = FAD, Null model = dk.

(d) Test = FAD, Null model = dk.

Fig. 3. Summary GP and FAD results on a grid of 33 modularity values in $[0, \ldots, 0.8]$, grid step 0.025, using dk as null model. In the first row the BFs from GP testing procedure are reported when using Fast Greedy (a) and Louvain (b) as community extraction methods. In the second row the p-values from the FAD testing procedure are reported when using Fast Greedy (c) and Louvain (d) as community extraction methods. The red line corresponds to 0.05 p-value. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

(see Sah et al., 2014 for details). Specifically, we generated a modular random graph on a grid of 33 modularity values $Q \in [0, \ldots, 0.8]$, grid step 0.025, using a power law for degree distribution and for module size distribution, with size = 2000, number of modules = 10 and average degree = 10. For each graph, the corresponding null model was generated using both the Configuration Model and the *dk* random model, as discussed in Section 3.

The application of the overall procedure on the simulated datasets is summarised in Figs. 2 and 3, respectively. In particular, Fig. 2 refers to the use of CM as null model and Fig. 3 refers to the use of *dk* as null model. In each figure we plot GP and Functional *Anderson-Darling* (*FAD*) test results versus the chosen grid of modularity values when extracting the communities either with *Fast Greedy* or *Louvain* methods.

Gaussian process results

The application of the Gaussian Processes approach described in Section 3.1 to the simulated networks is summarised in the first row ((a) and (b)) of Figs. 2 and 3. The resulting BFs plots show an overall growing trend from modularity $Q = 0$ to $Q = 0.8$ after clustering with either clustering *Fast Greedy* or *Louvain*. This gives strong statistical evidence that networks with a high modularity have a robust clustering structure (significantly different from the random). However, note that

Table 1
BF for the 5 *dcsbm* networks corresponding to $P_{in} = 0.3$ and $P_{out} = 0.08$.

nc	*Q*	*Sp*	*BF*
8	0.2202	0.0137	174.6640
10	0.1844	0.0105	124.7754
12	0.1663	0.0093	108.5694
14	0.1494	0.0087	57.8306
16	0.1359	0.0083	38.1446

Louvain method produces less oscillating results at low modularity ($Q \leq 0.2$) and very high modularity ($Q \geq 0.7$). On the other hand *Fast Greedy* produces more stable results at high modularity ($0.4 \leq Q \leq 0.6$), when using CM as null model. Both clustering methods induce a BF that is fast growing when $0.2 \leq Q \leq 0.4$ either when using CM or *dk* as null models.

Functional principal component testing results

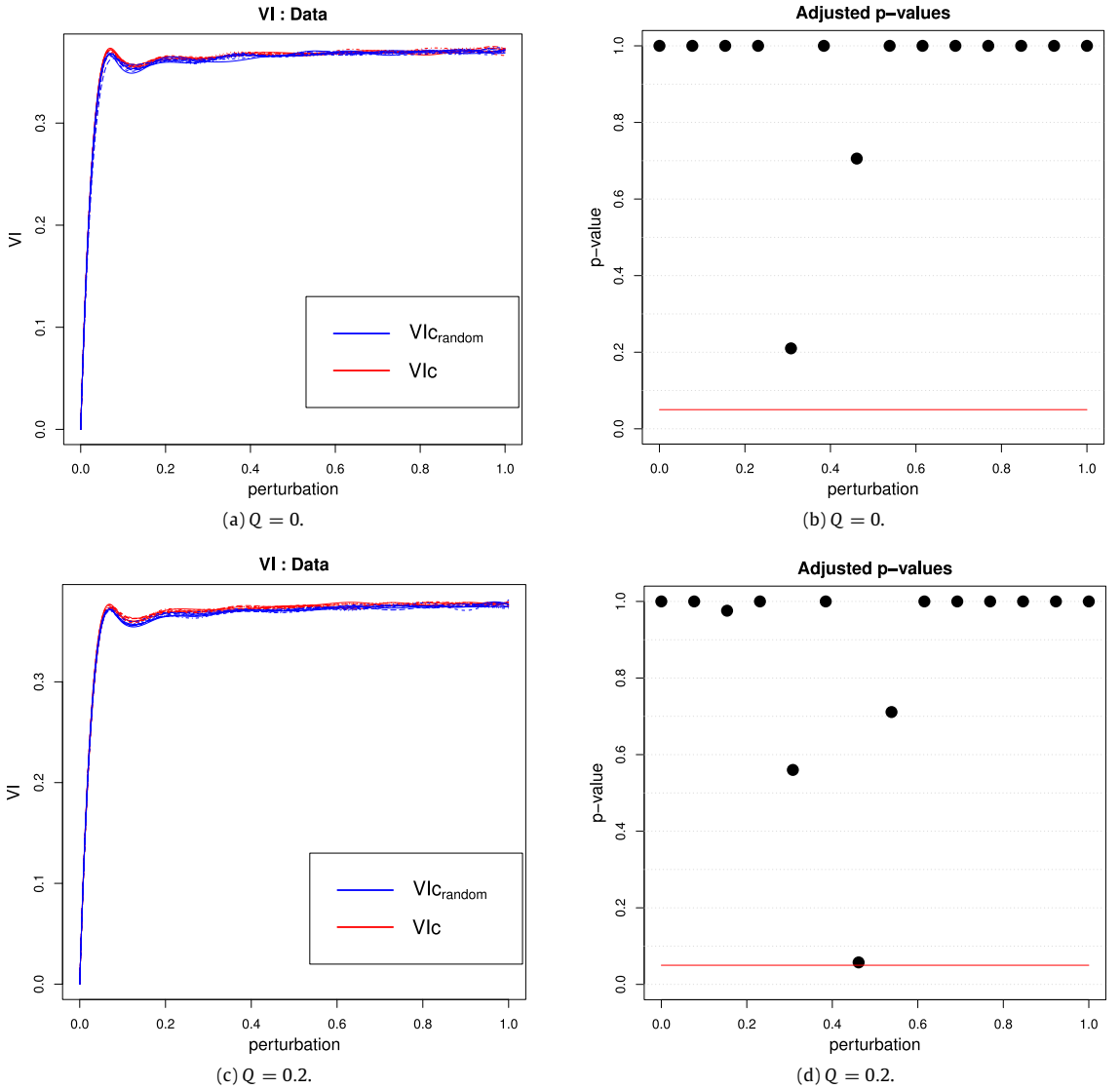
Similarly, the second row ((c) and (d)) of Figs. 2 and 3 summarise the application of the Functional *Anderson-Darling* test described in Section 3.2 to the simulated data. As we can see, the False Discovery rate adjusted *p*-values decreases drastically when the simulated network modularity grows from $Q = 0.275$ to $Q = 0.8$, for *Fast Greedy* and from $Q = 0.225$ to $Q = 0.8$ for *Louvain*, when using CM null model. Similarly, when using *dk* as null model, the adjusted *p*-values flatten under the significance value for $Q \geq 0.3$ (*Fast Greedy*) and for $Q \geq 0.35$ (*Louvain*). In the complementary intervals the result is oscillating around the significant threshold of 0.05, implying the incapability to clearly distinguish between the true and the random *VI* curves. This result agrees with the previous one obtained by GP.

Interval-wise functional testing results

The application of the Interval-wise Testing procedure described in Section 3.3 to the simulated datasets after clustering via *Fast Greedy* or *Louvain* are depicted respectively in Figs. 4 and 5. In this case we just show the outcomes on 5 simulated datasets with different modularity $Q \in [0, 0.2, 0.4, 0.6, 0.8]$, choosing CM as null model. In each figure, panels (a), (c), (e), (g), (i) show the VI curves for the null model (VIc_{random}) and for the actual model (VIc). The two curves appear to be very close for low modularity values and depart from each other as the modularity increases till a maximum of $Q = 1$. In panels (b), (d), (f), (h), (j) this is quantified locally by a specific adjusted *p*-value in each sub-interval. Significant *p*-values are falling under the horizontal red line corresponding to the critical value of 0.05. As we can see, either using *Louvain* or *Fast Greedy* as clustering methods yields to the similar *ITP* results conclusion. Of course when there is no perturbation (i.e. at level $p = 0$) the two curves are coincident and hence not significantly different. When $Q \geq 0.4$ the two *VI* curves are significantly different at any perturbation level, apart from some cases at $p \geq 0.5$ and $Q = 0.8$ where the VIc is close to the VIc_{random}, indeed note that if we strongly perturb a network ($p \geq 0.5$, i.e. we rewire more than 50% of edges) it approaches a random network. Also in this case *Louvain* is able to recover a non random clustering at lower modularity ($Q \geq 0.2$) than *Fast Greedy*, confirming the results obtained by the other two approaches. Moreover note that at $Q = 0$, when applying *Louvain*, the method detects some interval where there is a significant difference between the two curves but for strong perturbation level ($p \geq 0.5$).

4.2. Alternative data simulation strategy

In order to further explore the sensitivity of our approach with respect to the specific network structure, number of clusters and sparseness, we also applied it to random network graphs generated using the degree corrected stochastic block model (*dcsbm*). We implemented the approach proposed in Karrer and Newman (2011), where a *dcsbm* with closed-form parameter solutions is developed. This enabled us to generate networks with a specific number of communities, group assignment and edge probability distribution. In particular, for sake of simplicity, we simulated three different scenarios where we fixed the edge probability within each community P_{in} and the edge probability between each couple of communities P_{out}. In each scenario we used $P_{in} = 0.3$ and a number of vertex $nv = 2000$. In order to mimic broadly different modular structures, we generated the three scenario using $P_{out} = 0.08$, for a low modularity, $P_{out} = 0.03$, for a medium modularity, and $P_{out} = 0.008$, for a high modularity. For each scenario we then simulated five different networks corresponding to five different number of clusters *nc*, namely $nc = 8, 10, 12, 14, 16$, with a total of 15 networks. Each of the 15 networks was then analysed with our pipeline setting the *GP* as testing procedure, *CM* as null model and *Fast Greedy* as clustering method. For every generated network we also computed the true modularity values *Q* and the sparseness *Sp* defined in terms of percentage of edges $Sp = ne/(nv*(nv-1)/2)$, where *ne* is the total number of edges of the given network. The results for the three different scenarios are summarised in Tables 1–3 respectively. As you can see from Tables 1 and 2 the BF decreases with the number of clusters and increases with the modularity Q and the sparseness *Sp*, both at low modularity values ($P_{out} = 0.08$) and at medium modularity values ($P_{out} = 0.03$). More precisely Table 4 shows that the BF is highly negatively correlated with the *nc* and highly positively correlated with the modularity and sparsity values, both at low and medium modularity values. At the same time when the modularity is high ($P_{out} = 0.008$), the BF oscillates but is always very high ($BF \geq 268$) and hence the methodology is clearly able to detect the difference from the randomness at high modularity.



(a) $Q = 0$. (b) $Q = 0$.

(c) $Q = 0.2$. (d) $Q = 0.2$.

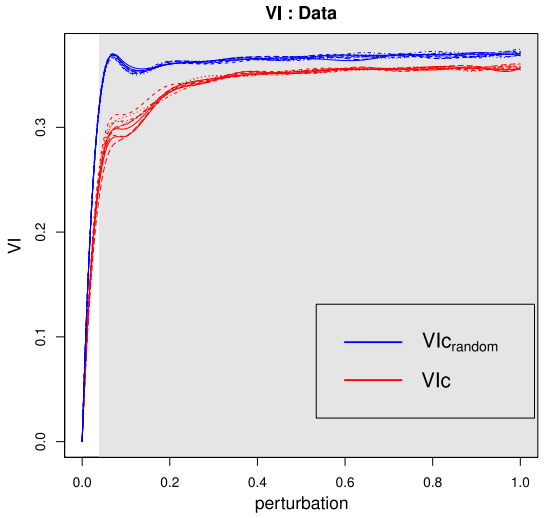
Fig. 4. VI plots on the clustering obtained via Fast Greedy on 5 simulated datasets with different modularity $Q \in [0, 0.2, 0.4, 0.6, 0.8]$ ($Q = 0$ (a), $Q = 0.2$ (c), $Q = 0.4$ (e), $Q = 0.6$ (g) and $Q = 0.8$ (i)) and corresponding adjusted p-values of the Interval Testing procedure ($Q = 0$ (b), $Q = 0.2$ (d), $Q = 0.4$ (f), $Q = 0.6$ (h) and $Q = 0.8$ (j)). Horizontal red line corresponds to the critical value 0.05. Light grey areas correspond to p-values below 0.05, dark grey areas correspond to p-values below 0.01. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Table 2
BF for the 5 *dcsbm* networks corresponding to $P_{in} = 0.3$ and $P_{out} = 0.03$.

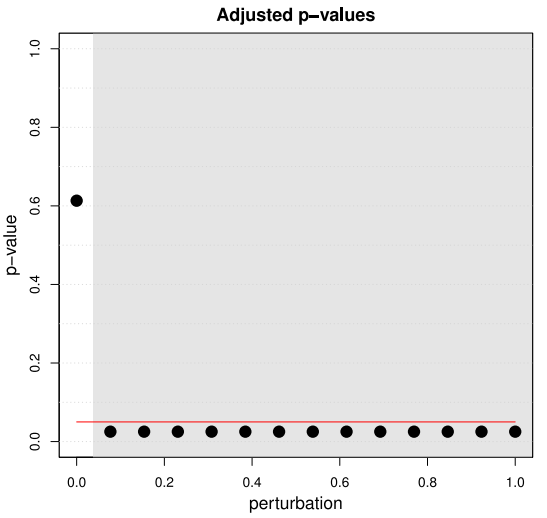
nc	*Q*	*Sp*	*BF*
8	0.4603	0.0108	249.7482
10	0.4177	0.0067	217.5520
12	0.3903	0.0053	232.6276
14	0.3648	0.0046	197.5130
16	0.3273	0.0041	128.9253

4.3. Application to real data

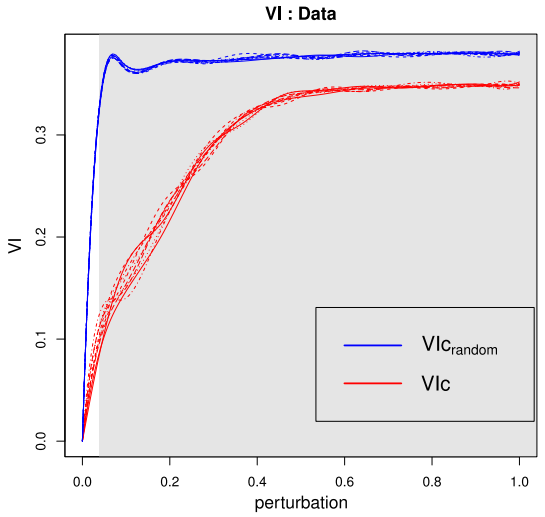
In order to provide an example of our analysis work-flow, we selected four different publicly available datasets namely two biological (protein–protein interaction networks, *Nexus* 5 and *Barabasi*), one representing the western states power grid



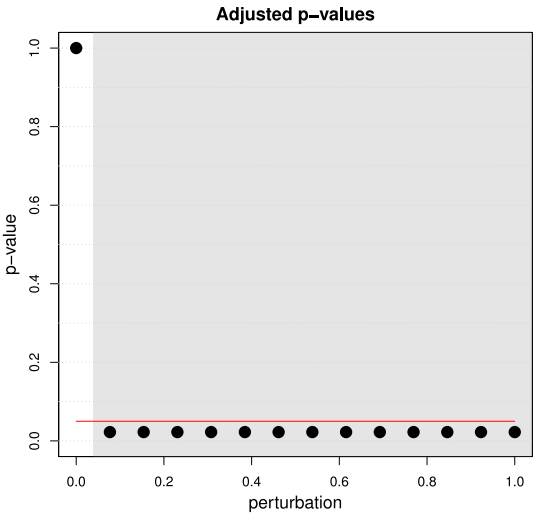
(e) $Q = 0.4$.



(f) $Q = 0.4$.



(g) $Q = 0.6$.



(h) $Q = 0.6$.

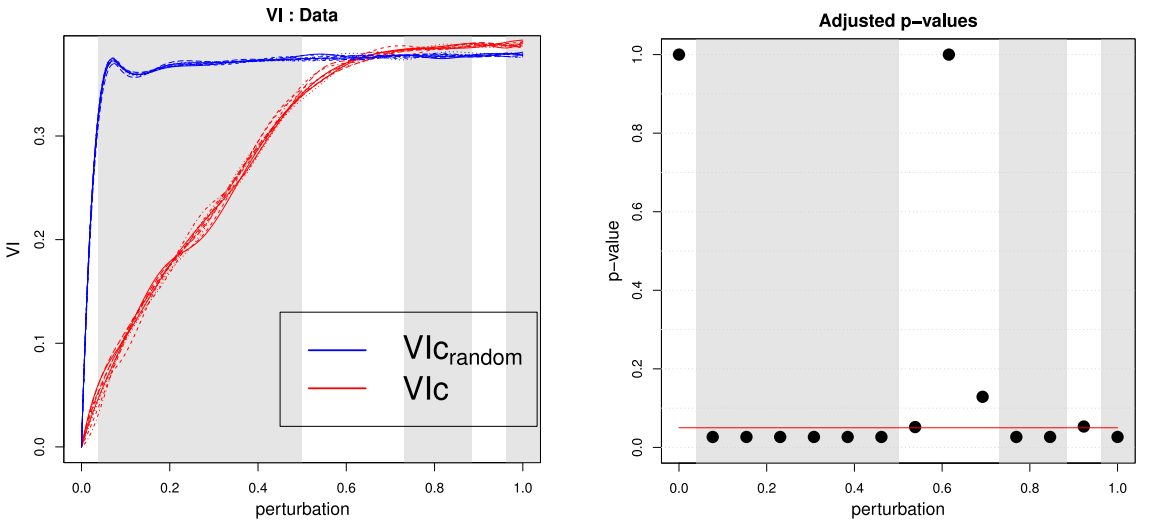
Fig. 4. (*continued*)

Table 3
BF for the 5 *dcsbm* networks corresponding to $P_{in} = 0.3$ and $P_{out} = 0.008$.

nc	*Q*	*Sp*	*BF*
8	0.7122	0.0095	332.0044
10	0.7089	0.0050	270.6336
12	0.6871	0.0036	358.0979
14	0.6657	0.0028	340.2443
16	0.6294	0.0023	268.1435

Table 4
Correlation coefficient between the BF and the *nc*, *Q* and *Sp* values for the three different scenarios simulated with the *dcsbm* model.

P_{out}	Corr(BF, nc)	Corr(BF, Q)	Corr(BF, Sp)
0.08	−0.987	0.984	0.931
0.03	−0.883	0.891	0.709
0.008	−0.220	0.277	0.175



(i) $Q = 0.8$. (j) $Q = 0.8$.

Fig. 4. (*continued*)

Table 5
Number of vertex (nv) and number of edges (ne) for each of the 4 real analysed datasets and for the two less modular ($Q = 0.29$) and the most modular ($Q = 0.68$) ego networks in Facebook (FB).

Real networks summary						
	FB	FB ($Q = 0.29$)	FB ($Q = 0.68$)	Nexus 5	Nexus 15	Barabasi
nv	4039	224	534	2617	4941	1870
ne	88234	3192	4813	11855	6594	2240

of United States (*Nexus* 15) and a social dataset (*Facebook*). Note that *Nexus* is an online repository of networks, with an *API* that allow programmatic queries against it, and programmatic data download as well. These functions can be used to query it and download data from it, directly as an *igraph* graph. The total number of nodes and edges of these four real networks are summarised in Table 5 and displayed in Fig. 6.

Nexus 5

This dataset consists of an undirected protein–protein interaction network in yeast. This dataset was compiled by von Mering et al. (see von Mering et al., 2002) combining various sources. Only the interactions that have high and medium confidence are included here.

Protein–protein interaction (Barabasi)

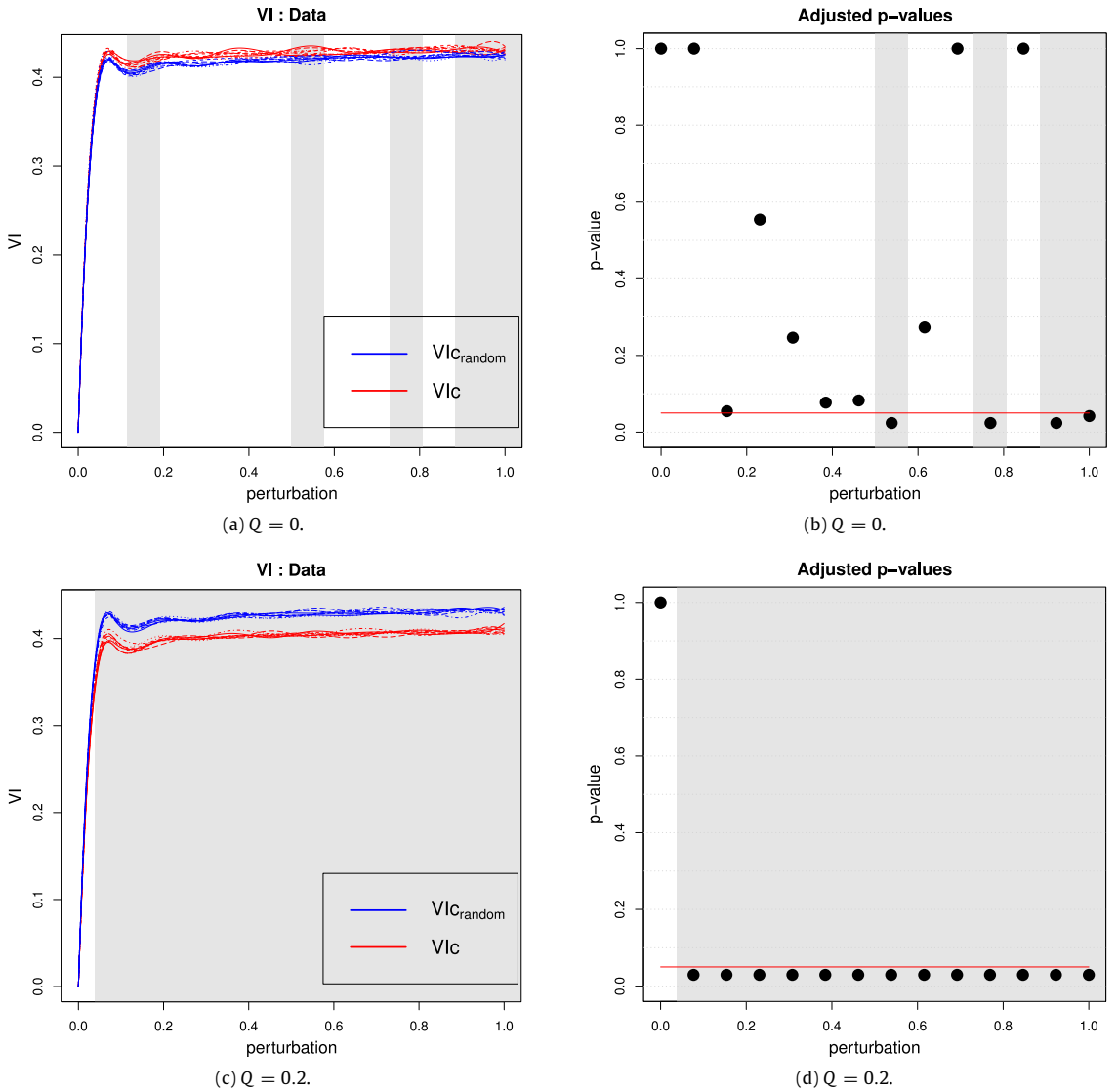
This dataset consists of the protein–protein interaction network in Saccharomyces cerevisiae described and analysed in Jeong et al. (2001). It is derived from combined, non-overlapping data, obtained mostly by systematic two-hybrid analyses. Data are available at http://www3.nd.edu/~networks/resources.htm.

Nexus 15

This dataset is an undirected, unweighted network representing the topology of the Western States Power Grid of the United States and was compiled by Duncan Watts and Steven Strogatz. Data are available at http://cdg.columbia.edu/cdg/datasets, Watts and Strogatz (1998).

Facebook

This dataset consists of 'circles' (or 'friends lists') from Facebook (McAuley and Leskovec, 2012). The authors obtained profile and network data from 10 ego-networks, consisting of 193 circles and 4039 users. To do so they developed their own Facebook application and conducted a survey of ten users, who were asked to manually identify all the circles to which their friends belonged. On average, users identified 19 circles in their ego-networks, with an average circle size of



(a) $Q = 0$. (b) $Q = 0$.

(c) $Q = 0.2$. (d) $Q = 0.2$.

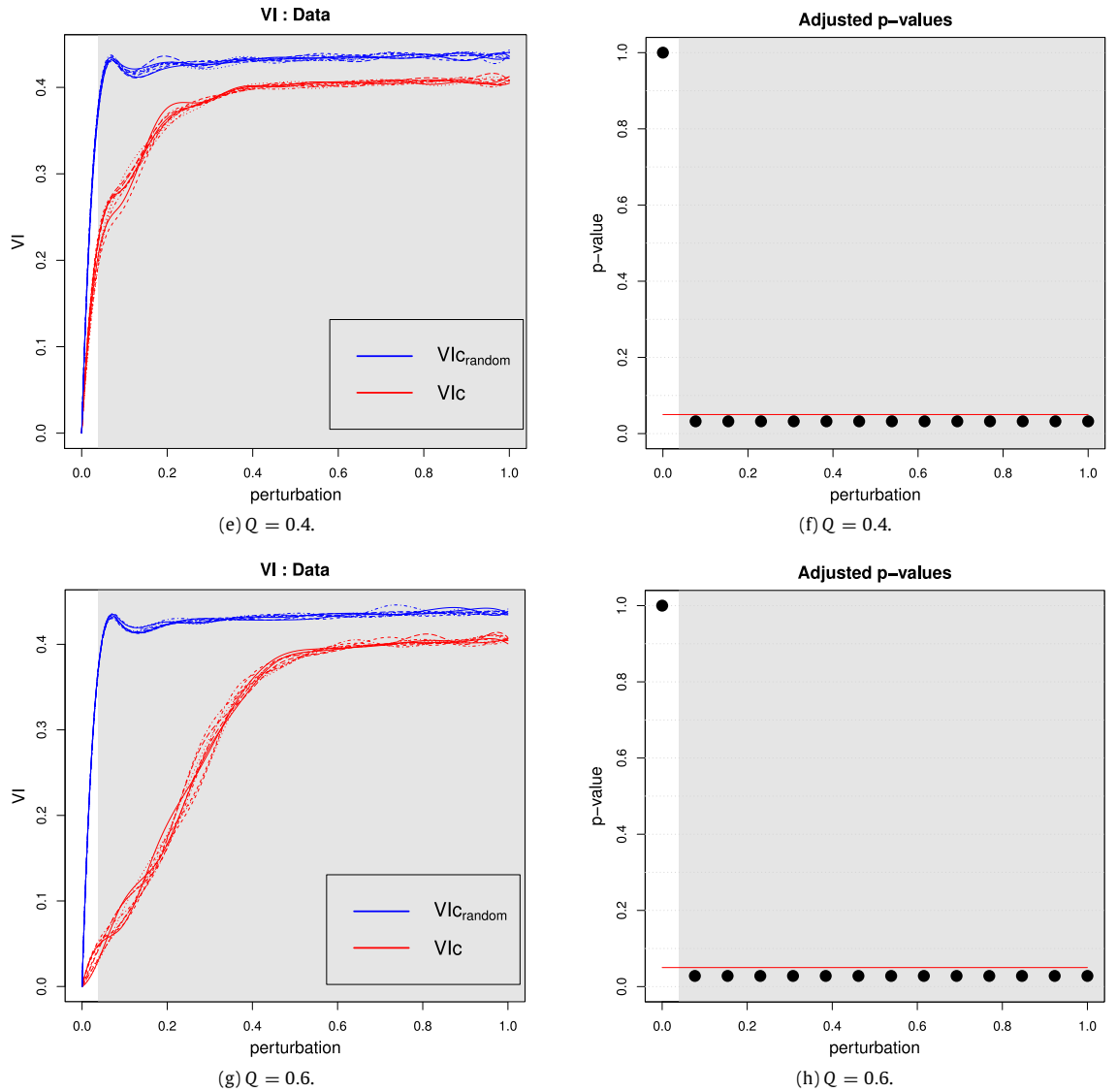
Fig. 5. VI plots on the clustering obtained via Louvain on 5 simulated datasets with different modularity $Q \in [0, 0.2, 0.4, 0.6, 0.8]$ ($Q = 0$ (a), $Q = 0.2$ (c), $Q = 0.4$ (e), $Q = 0.6$ (g) and $Q = 0.8$ (i)) and corresponding adjusted p-values of the Interval Testing procedure ($Q = 0$ (b), $Q = 0.2$ (d), $Q = 0.4$ (f), $Q = 0.6$ (h) and $Q = 0.8$ (j)). Horizontal red line corresponds to the critical value 0.05. Light grey areas correspond to p-values below 0.05, dark grey areas correspond to p-values below 0.01. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

22 friends. Examples of such circles include students of common universities, sports teams, relatives, etc. Data are available at http://snap.stanford.edu/data/egonets-Facebook.html. In particular we analysed: the overall 10 ego-networks as a single network, the less modular ($Q = 0.29$) and the most modular ($Q = 0.68$) ego network separately. In this case the modularity was computed according to the *Fast Greedy* partition. This case study shows us how our results could be used to compare and understand different network structures.

The application of the overall procedure on the just described real datasets is summarised in Tables 6 and 7 and in Figs. 8 and 9, respectively. The two single ego-networks corresponding to the less modular ($Q = 0.29$) and the most modular ($Q = 0.68$), are plotted in Fig. 7.

Gaussian process results

The application of the Gaussian Processes approach described in Section 3.1 to the four real networks is summarised in Table 6. The resulting BF are very high for either *Fast Greedy* or *Louvain* clustering. This gives strong statistical evidence that



(e) $Q = 0.4$. (f) $Q = 0.4$.

(g) $Q = 0.6$. (h) $Q = 0.6$.

Fig. 5. (*continued*)

the four analysed networks have a robust clustering structure hence the recovered community structures are not likely to be random.

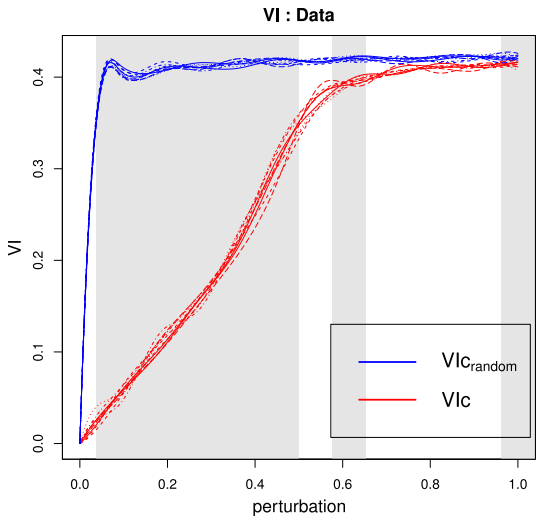
Functional principal component testing results

Similarly, Table 7 summarises the application of the Functional *Anderson-Darling* test described in Section 3.2 to the four real networks. As we can see the False Discovery rate adjusted p-values are well lower then the standard significance value 0.05, after clustering with either clustering *Fast Greedy* or *Louvain*. This result agree with the previous one leading to the same conclusion that analysed real networks have a robust clustering structure.

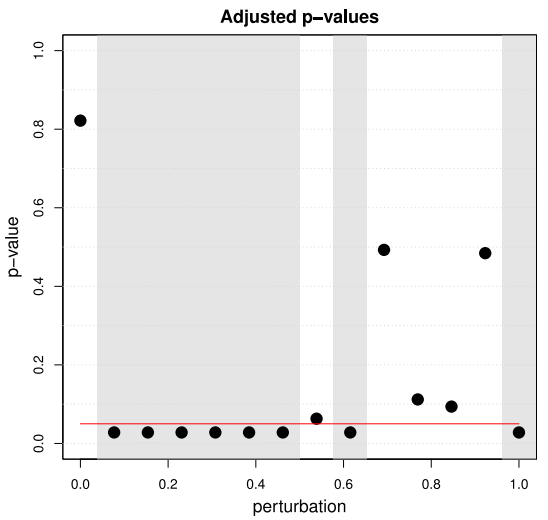
Interval-wise functional testing results

The application of the Interval-wise Testing procedure described in Section 3.3 to the real datasets after clustering via *Fast Greedy* or *Louvain* are depicted respectively in Figs. 8 and 9.

In each figure, panels (a), (c), (e), (g) show the VI curves for the null model (VIc_{random}) and for the actual model (VIc). In all the cases the two curves appear to be very close for high perturbation values and depart from each other as perturbation level approaches zero. In panels (b), (d), (f), (h) this is quantified locally by a specific adjusted p-value in each sub-interval. Also in this case significant p-values are falling under the horizontal red line corresponding to the critical value of 0.05. As

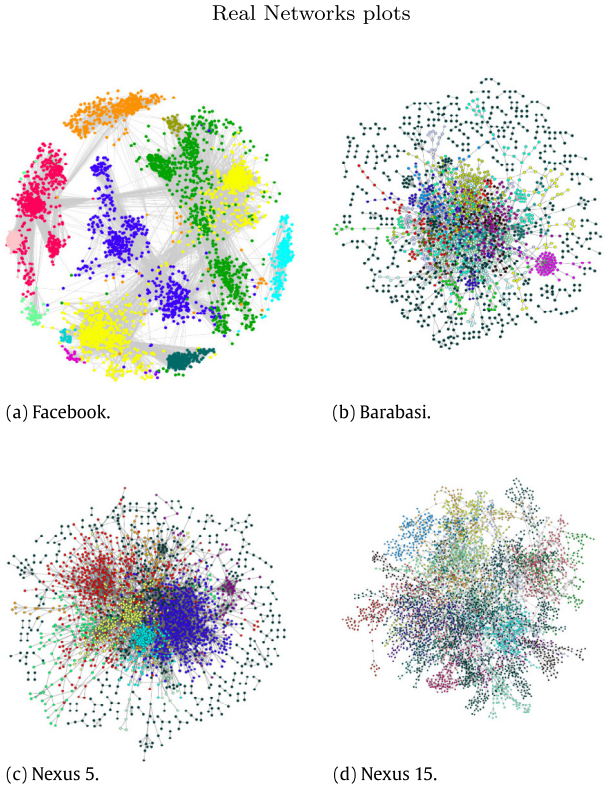


(i) $Q = 0.8$.



(j) $Q = 0.8$.

Fig. 5. (*continued*)



(a) Facebook.

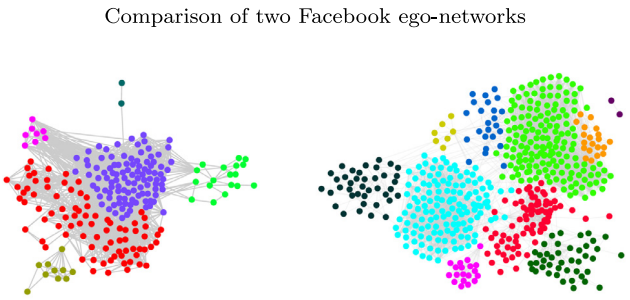
(b) Barabasi.

(c) Nexus 5.

(d) Nexus 15.

Fig. 6. For each real network (Facebook (a), Barabasi (b), Nexus 5 (c) and Nexus 15 (d)) we show the extracted community found by the proposed method *Fast Greedy*. Only the community with more than the 5% of nodes is displayed.

expected, either using *Louvain* or *Fast Greedy* as clustering methods yields to similar results conclusion. As already observed for the synthetic datasets, if we strongly perturb a network ($p \geq 0.5$, i.e. we rewire more than 50% of edges) it approaches a random network, indeed the two *VI* curves become very close, and the *p*-value could survive the threshold.



(a) Q = 0.29 Facebook ego-network. (b) Q = 0.68 Facebook ego-network.

Fig. 7. For both the less modular (a) and the most modular (b) ego networks, we show the extracted community found by the proposed method *Fast Greedy*. Only the community with more than the 5% of nodes is displayed.

Table 6
GP Bayes Factor on the 4 datasets and for the two less modular ($Q = 0.29$) and the most modular ($Q = 0.68$) ego networks in Facebook (*FB*), after clustering via *Fast Greedy* or *Louvain*.

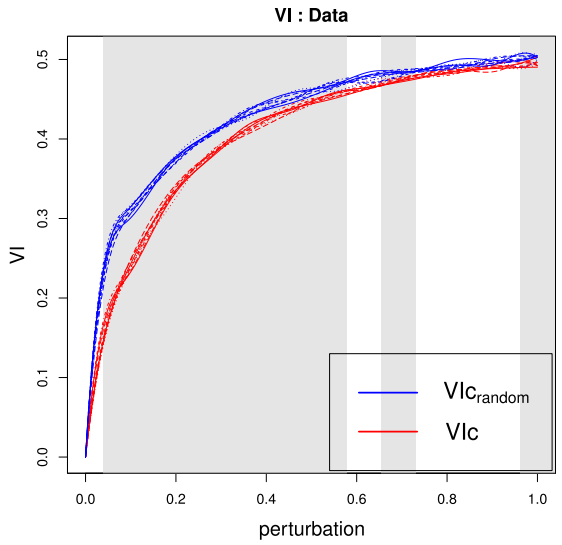
Datasets	Fast Greedy	Louvain
Barabasi	284.411	243.816
FB (10 ego)	297.251	361.060
FB ($Q = 0.29$)	153.760	258.920
FB ($Q = 0.68$)	290.503	338.269
Nexus 5	340.795	431.477
Nexus 15	503.183	495.810

Table 7
FDR adjusted p-values by the FAD test procedure on the 4 datasets and for the two less modular ($Q = 0.29$) and the most modular ($Q = 0.68$) ego networks in Facebook (*FB*), after clustering via *Fast Greedy* or *Louvain*.

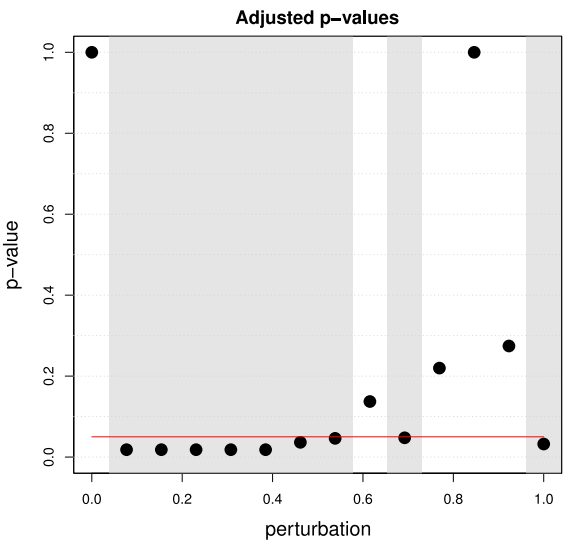
Datsets	Fast Greedy	Louvain
Barabasi	0.00016	0.00302
FB (10 ego)	0.00024	8e−05
FB ($Q = 0.29$)	0.00016	0.0155
FB ($Q = 0.68$)	8e−05	8e−05
Nexus 5	0.000765	0.00024
Nexus 15	8e−05	8e−05

5. Metrics comparison

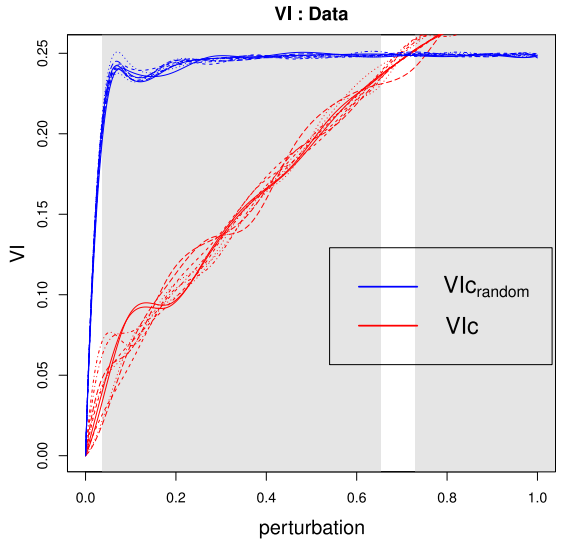
The crucial step of our proposal relies on the choice of the best testing procedure, given the best clustering. Following the simulation study results described in Section 4.1, we can conclude that there is not an absolute best solution, but the outcome is intrinsically linked to the structural properties of the network under study. The most relevant observed dependence is from the modularity. However the value of modularity is mainly unknown in real networks and it has to be estimated via a community extraction method. This is already a potential first bias as the clustering method chosen might be not the optimal one for the network under study. For an insightful discussion about the choice of the community extraction method, we defer to a recent work Yang et al. (2016) that carefully address this problem. However the most reliable outcome, in terms of stability and interpretability, is obtained by the *GP* testing on the *VI* curves, using *CM* as null random model. This is pointed out in Section 4.1 where the simulation study is performed. Indeed there is a high positive correlation between the *BF* and the modularity values Q. In Fig. 2 shows an overall BF growing trend from modularity $Q = 0$ to $Q = 0.8$ after clustering with either *Fast Greedy* or *Louvain* methods. This supports the assumption that networks with a high modularity have a community structure that is significantly different from the random. Both clustering methods induce a BF that is fast growing when $0.2 \leq Q \leq 0.4$, very high and more stable at high modularity and very low and stable at low modularity. Furthermore in Section 4.2 and in Table 4 we show that the BF is highly negatively correlated with the number of clusters and highly positively correlated with the modularity and sparsity values at low and medium modularity values, hence confirming that the GP testing can help to discriminate not only between networks but also between the specific network structures. As for the Functional Principal Component testing results, as shown in 4.1, the p-values are oscillating at low modularities while very low otherwise. This is result is less interpretable and does not allow to discriminate between networks according



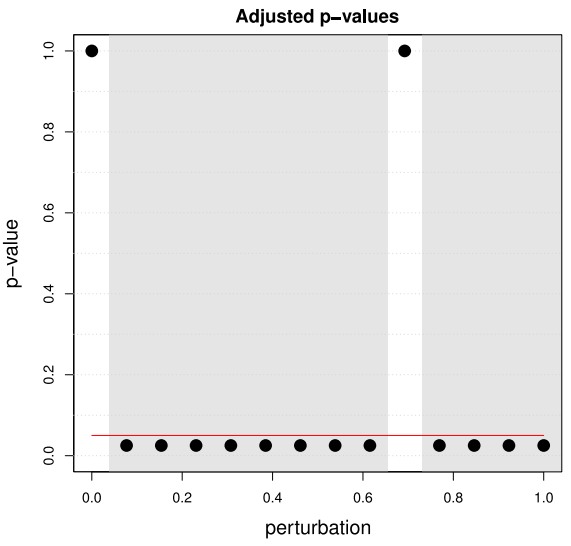
(a) Barabasi.



(b) Barabasi.



(c) Facebook.



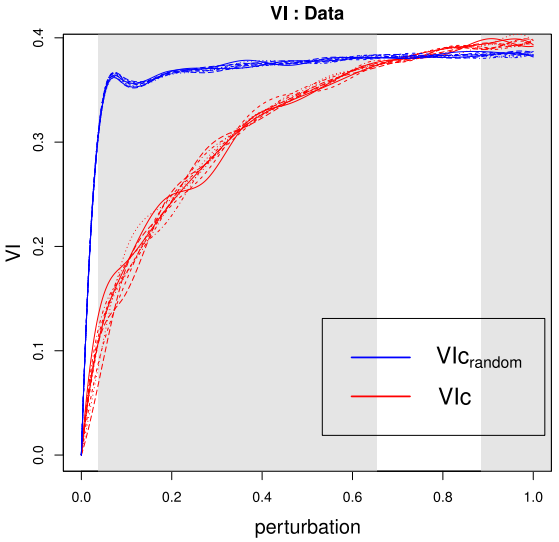
(d) Facebook.

Fig. 8. VI plots on the clustering obtained via Fast Greedy on real datasets (Barabasi (a), Facebook (c), Nexus 5 (e) and Nexus 15 (g)) and the corresponding adjusted p-values of the Interval Testing procedure (Barabasi (b), Facebook (d), Nexus 5 (f) and Nexus 15 (h)). Horizontal red line corresponds to the critical value 0.05. Light grey areas correspond to p-values below 0.05, dark grey areas correspond to p-values below 0.01. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

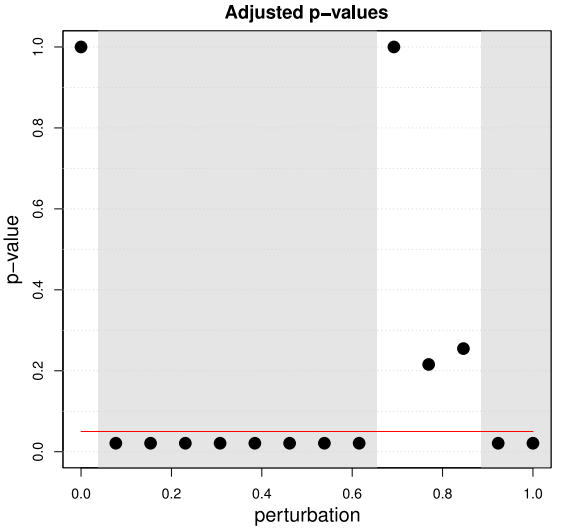
to their properties. Finally the Interval-wise Functional Testing produces interesting results at specific modularity intervals, hence this could be a valid method when a specific modularity interval is of interest.

6. Conclusions and discussions

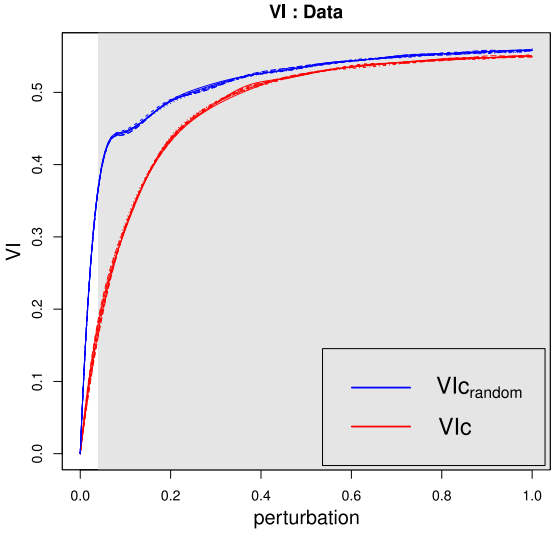
In this paper we propose an effective procedure to evaluate the robustness of a clustering. Given a community detection method and a network of interest, our methodology enables to clearly detect if the community structure found by some algorithms is statistically significant or is a result of chance, permitting to examine the stability of the partition recovered. As suggested in Karrer et al. (2008), we specify a perturbation strategy and a null model to build a set of procedures based on VI as stability measure. This enables to build the VI curve as a function of the perturbation percentage and to compare it with the corresponding null model curve in the functional data analysis framework.



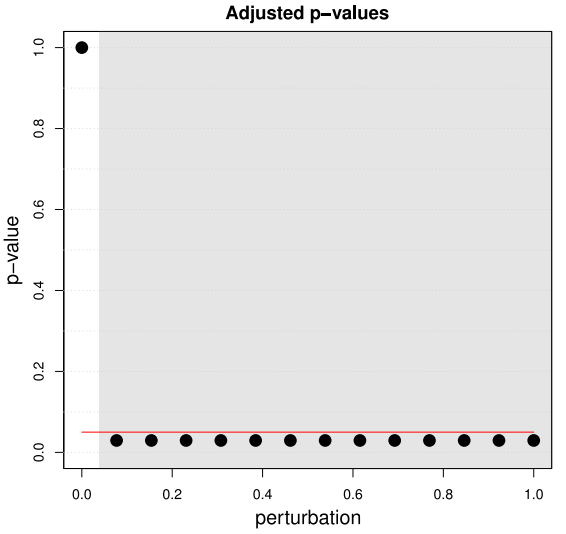
(e) Nexus 5.



(f) Nexus 5.



(g) Nexus 15.



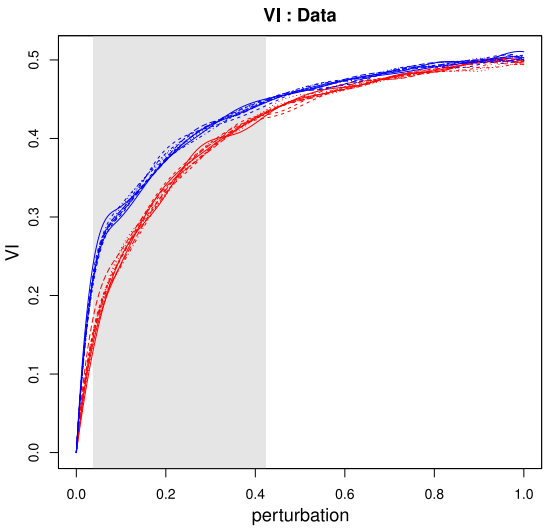
(h) Nexus 15.

Fig. 8. (*continued*)

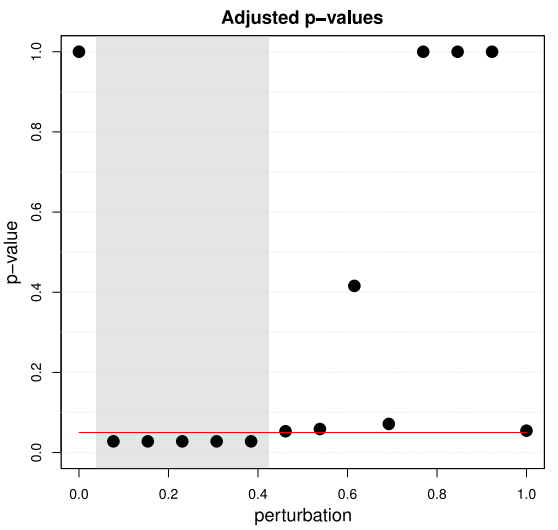
We point out that our methodology could also be used to compare different clustering methodologies, indeed given two clusterings on the same network, we could test the agreement between the two recovered partitions via the direct comparison of the corresponding VI curves as defined by our procedure in Section 3. For example, note that *Louvain* method is able to recover a non random clustering also at low modularity ($Q \geq 0.2$), while *Fast Greedy* is able to recover a non random clustering a for modularities $Q \geq 0.3$. This indicates that perhaps *Louvain* is more suited for networks having a weak community structure.

However, it is out of the scope of the present paper the comparative evaluation of different community extraction methods. The two methodologies *Louvain* and *Fast Greedy* were indeed only instrumental to the exemplification of our procedure. Both of them were selected at this stage as they both enables for an automatic definition of the optimal number of communities and are based on the optimisation of the modularity, that plays a key role in describing community structures.

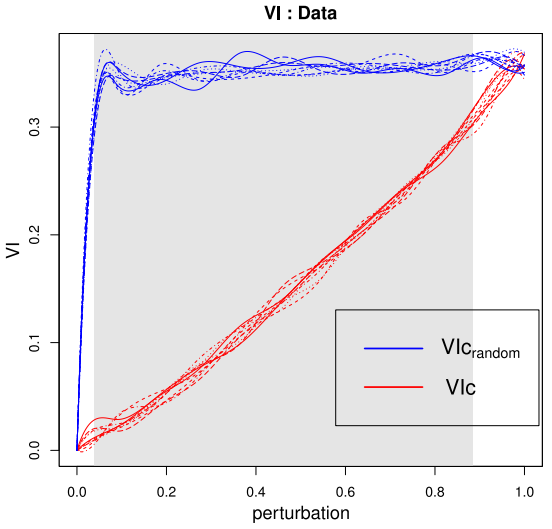
As a general conclusion, as highlighted in Section 5, the most reliable outcome, in terms of stability and interpretability, is obtained by the *GP* testing on the *VI* curves, using *CM* as null random model.



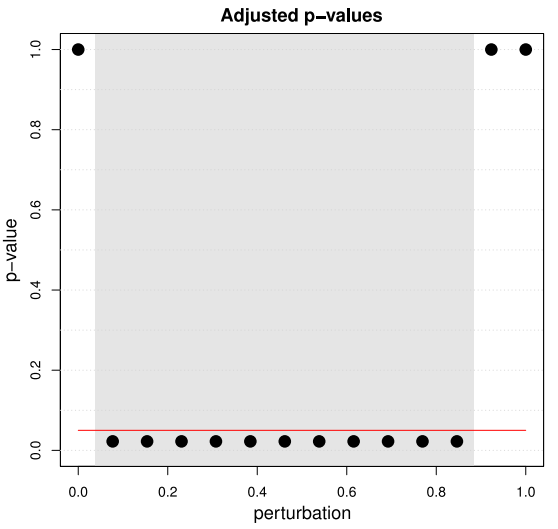
(a) Barabasi.



(b) Barabasi.



(c) Facebook.



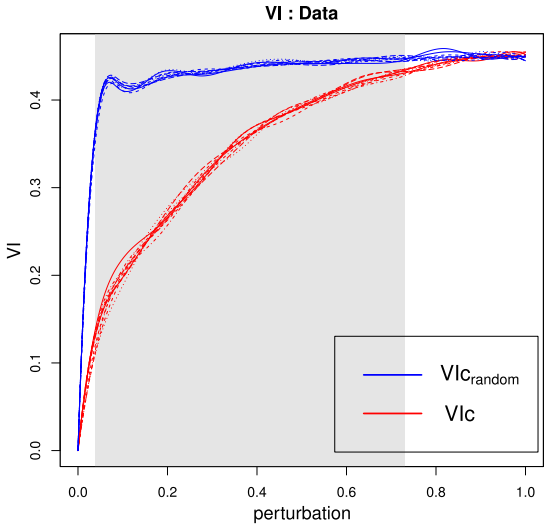
(d) Facebook.

Fig. 9. VI plots on the clustering obtained via Louvain on real datasets (Barabasi (a), Facebook (c), Nexus 5 (e) and Nexus 15 (g)) and the corresponding adjusted p-values of the Interval Testing procedure (Barabasi (b), Facebook (d), Nexus 5 (f) and Nexus 15 (h)). Horizontal red line corresponds to the critical value 0.05. Light grey areas correspond to p-values below 0.05, dark grey areas correspond to p-values below 0.01. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

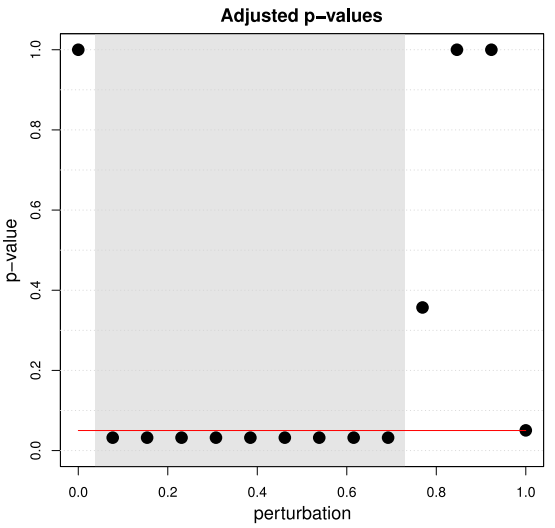
An interesting and straightforward extension of the current paper would be using a different clustering stability measure, for example the *Normalized Mutual Information* measure proposed in Danon et al. (2005). This would also lead to a comparison of the performance of different measures for community structure comparison.

Acknowledgement

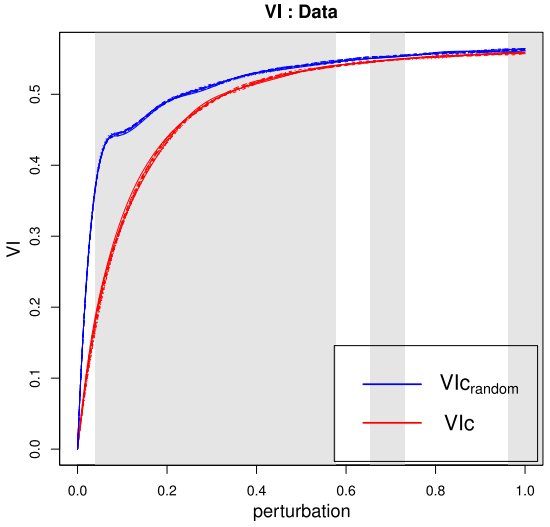
The authors equally contributed to the paper. The work of Luisa Cutillo has been supported by the European Union under Horizon 2020, Marie Sklodowska-Curie Individual Fellowship (EU project CONTESSA — H2020-MSCA-IF-2014 number 660388).



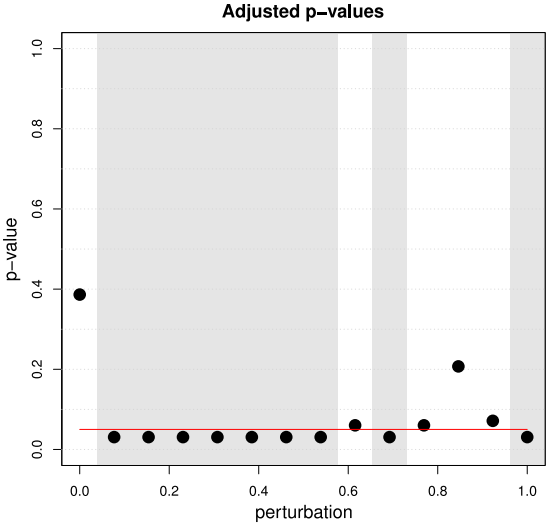
(e) Nexus 5.



(f) Nexus 5.



(g) Nexus 15.



(h) Nexus 15.

Fig. 9. (*continued*)

References

Bender, E.A., Canfield, E.R., 1978. The asymptotic number of labeled graphs with given degree sequences. J. Combin. Theory A 24, 296–307 MR0505796.
Benjamini, Y., Hochberg, Y., 1995. Controlling the false discovery rate: a practical powerful approach to multiple hypothesis testing. J. R. Stat. Soc. Ser. B Stat. Methodol. 57, 289–300.
Bianconi, G., Pin, P., Marsili, M., 2009. Assessing the relevance of node features for network structure. Proc. Natl. Acad. Sci. 106, 11433.
Blondel, A.V., Guillaume, J.L., Lambiotte, R., Lefebvre, E., 2008. Fast unfolding of communities in large networks. J. Stat. Mech. P10008.
Clauset, A., Newman, M.E.J., Moore, C., 2004. Finding community structure in very large networks. Phys. Rev. E 70, 066111.
Coscia, M., Giannotti, F., Pedreschi, D., 2011. A classification for community discovery methods in complex networks. Stat. Anal. Data Min. 4, 512–546 MR2842407.
Cutillo, L., Carissimo, A., Figini, S., 2012. Network selection: A method for ranked lists selection. PLoS One 7 (8), e43678.
Danon, L., Guilera, A.D., Duch, J., Arenas, A., 2005. Comparing community structure identification. J. Stat. Mech.: Theory Exp. 2005.
De Vico Fallani, F., Nicosia, V., Latora, V., Chavez, M., 2014. Nonparametric resampling of random walks for spectral network clustering. Phys. Rev. E 89, 012802.
Fortunato, S., 2010. Community detection in graphs. Phys. Rep. 486, 75–174.
Gel, Y.R., Lyubchich, V., Ramirez Ramirez, L.L., 2017. Bootstrap quantification of estimation uncertainties in network degree distributions. Sci. Rep. 7, 5807.

Gfeller, D., Chappelier, J.C., De Los Rios, P., 2005. Finding instabilities in the community structure of complex networks. Phys. Rev. E 72, 056135.
Gkantsidis, C., Mihail, M., Zegura, E., 2003. The Markov chain simulation method for generating connected power law random graphs. In: Proceedings of the Fifth Workshop on Algorithm Engineering and Experiments. SIAM, pp. 16–25.
Goldenberg, A., Zheng, A.X., SE Fienberg, S.E., Airoldi, E.M., 2010. A survey of statistical network models. Found. Trends Mach. Learn. 2, 129–233.
Guimera, R., Sales-Pardo, M., Amaral, L.A.N., 2004. Modularity from fluctuations in random graphs and complex networks. Phys. Rev. E 70, 025101.
Hakimi, S.L., 1962. On realizability of a set of integers as degrees of the vertices of a linear graph. I. J. Soc. Ind. Appl. Math. 10 (3), 496–506.
Harenberg, S., Bello, G., Gjeltema, L., Ranshous, S., Harlalka, J., Seay, R., Padmanabhan, K., Samatova, N., 2014. Community detection in large-scale networks: a survey and empirical evaluation. WIREs Comput. Stat. 6, 426–439.
Havel, V., 1955. A remark on the existence of finite graphs. Cas. Pest. Mat. 80, 477–480.
Jeong, H., Mason, S.P., Barabási, A.-L., Oltvai, Z.N., 2001. Lethality and centrality in protein networks. Nature 411, 41–42.
Kalaitzis, A.A., Lawrence, N.D., 2011. A simple approach to ranking differentially expressed gene expression time courses through Gaussian process regression. BMC Bioinformatics 12, 180.
Karrer, B., Levina, E., Newman, M.E.J., 2008. Robustness of community structure in networks. Phys. Rev. E 77, 046119.
Karrer, Brian, Newman, M.E.J., 2011. Stochastic blockmodels and community structure in networks. Phys. Rev. E 83 (1), 016107.
Kolacyzk, E.D., 2009. Statistical Analysis of Network Models. Springer, New York.
Lancichinetti, A., Radicchi, F., Ramasco, J.J., 2010. Statistical significance of communities in networks. Phys. Rev. E 81, 046110.
Lancichinetti, A., Radicchi, F., Ramasco, J.J., Fortunato, S., 2011. Finding statistically significant communities in networks. PLoS One 6 (4), e18961.
Massen, C.P., Doye, J.P.K., 2006. Thermodynamics of Community Structure. arXiv:cond-mat/0610077v1.
McAuley, J., Leskovec, J., 2012. Learning to Discover Social Circles in Ego Networks. NIPS. pp. 548–556.
Meilă, M., 2007. Comparing clusterings–an information based distance. J. Multivariate Anal. 98, 873–895.
Newman, M.E.J., 2003. The structure and function of complex networks. SIAM Rev. 45 (2), 167–256.
Newman, M.E.J., Girvan, M., 2004. Finding and evaluating community structure in networks. Phys. Rev. E 69, 026113.
Orsini, C., et al., 2015. Qunatifying randomness in real networks. Nat. Commun. 6, 8627.
Pesarin, F., Salmaso, L., 2010. Permutation Tests for Complex Data: Theory, Applications and Software. John Wiley & Sons Inc., Chichester.
Petit, A.N., 1976. A two-sample Anderson-Darling rank statistic. Biometrics 63, 161–168.
Pini, A., Vantini, S., 2016. The interval testing procedure: A general framework for inference in functional data analysis. Biometrics. http://dx.doi.org/10.1111/biom.12476.
Pomann, G.-M., Staicu, A.-M., Ghosh, S., 2016. A two-sample distribution-free test for functional data with application to a diffusion tensor imaging study of multiple sclerosis. J. Roy. Statist. Soc.: Ser. C 65, 395–414. http://dx.doi.org/10.1111/rssc.12130.
Porter, M.A., Onnela, J.-P., Mucha, P.J., 2009. Communities in networks. Notices Amer. Math. Soc. 56, 1082–1097 MR2568495.
Ramsay, J.O., Silverman, B.W., 1997. Functional Data Analysis. In: Springer Series in Statistics, ISBN: 978-1-4757-7109-1, (Print) 978-1-4757-7107-7 (Online).
Ramsay, J.O., Silverman, B.W., 2002. Applied Functional Data Analysis: Methods and Case Studies. In: Springer Series in Statistics, ISBN: 978-0-3872-2465-7, (online) - 978-0-3879-5414-1.
Rasmussen, C.E., Williams, C.K.I., 2006. Gaussian Processes for Machine Learning. The MIT Press, ISBN: 0-262-18253-X.
Reichardt, J., Bornholdt, S., 2007. Partitioning and modularity of graphs with arbitrary degree distribution. Phys. Rev. E 76, 015102.
Rosvall, M., Bergstrom, C., 2010. Mapping change in large networks. PLoS One 5 (1), e8694.
Sah, P., Singh, L.O., Clauset, A., Bansal, S., 2014. Exploring community structure in biological networks with random graphs. BMC Bioinformatics 15, 220. http://dx.doi.org/10.1186/1471-2105-15-220.
von Mering, C., Krause, R., Snel, B., Cornell, M., Oliver, S.G., Fields, S., Bork, P., 2002. Comparative assessment of large-scale data sets of protein-protein interactions. Nature 417, 399–403.
Wade, S., Ghahrmani, Z., 2015. Bayesian cluster analysis: Point estimation and credible balls. arXiv:1505.03339.
Watts, D.J., Strogatz, S.H., 1998. Collective dynamics of 'small-world' networks. Nature 393, 440–442.
Wilson, J.D., Wang, S., Mucha, P.J., Bhamidi, S., Nobel, A.B., 2014. A testing based extraction algorithm for identifying significant communities in networks. Ann. Appl. Stat. 8 (3), 1853–1891 MR3271356.
Yang, Z., Algesheimer, R., Tessone, C.J., 2016. A comparative analysis of community detection algorithms on artificial networks. Nat. Commun. 6, 30750.
Zweig, K.A., 2016. Network Analysis Literacy - A Practical Approach to the Analysis of Networks. Springer-Verlag, Wien.