

Community Detection for Correlation Matrices

Mel MacMahon¹ and Diego Garlaschelli^{1,2}

¹*Lorentz Institute for Theoretical Physics, University of Leiden,
Niels Bohrweg 2, 2333 CA Leiden, The Netherlands*

²*CABDyN Complexity Centre, Saïd Business School, University of Oxford,
Park End Street, OX1 1HP Oxford, United Kingdom*

(Received 16 November 2013; revised manuscript received 15 April 2014; published 14 April 2015)

A challenging problem in the study of complex systems is that of resolving, without prior information, the emergent, mesoscopic organization determined by groups of units whose dynamical activity is more strongly correlated internally than with the rest of the system. The existing techniques to filter correlations are not explicitly oriented towards identifying such modules and can suffer from an unavoidable information loss. A promising alternative is that of employing community detection techniques developed in network theory. Unfortunately, this approach has focused predominantly on replacing network data with correlation matrices, a procedure that we show to be intrinsically biased because of its inconsistency with the null hypotheses underlying the existing algorithms. Here, we introduce, via a consistent redefinition of null models based on random matrix theory, the appropriate correlation-based counterparts of the most popular community detection techniques. Our methods can filter out both unit-specific noise and system-wide dependencies, and the resulting communities are internally correlated and mutually anticorrelated. We also implement multiresolution and multifrequency approaches revealing hierarchically nested subcommunities with “hard” cores and “soft” peripheries. We apply our techniques to several financial time series and identify mesoscopic groups of stocks which are irreducible to a standard, sectorial taxonomy; detect “soft stocks” that alternate between communities; and discuss implications for portfolio optimization and risk management.

DOI: [10.1103/PhysRevX.5.021006](https://doi.org/10.1103/PhysRevX.5.021006)

Subject Areas: Complex Systems,
Interdisciplinary Physics,
Statistical Physics

I. INTRODUCTION

Over the past couple of decades, the amount of raw data available has started to grow at an exponential rate, doubling approximately every 12 months [1], while the amount of data being consumed by users remains linear [2]. The so-called “big data” phenomenon imposes an urgent need to develop, possibly with the aid of high-speed computing and cheap data storage, efficient pattern-detection methods and data-mining techniques aimed at identifying a few highly relevant pieces of information in an ever-increasing noisy or irrelevant background. One of the most important and widespread examples of the big data phenomenon is time-series data, as witnessed by the impressive growth of data bases of electronic and mobile-device communication patterns in large social systems, financial returns in stock markets, physiological signals such as heartbeat and brain dynamics, gene expression profiles, and finally, climate, weather, and earthquake activity. In all these examples, high-dimensional

(multiple) time series originate from the dynamical activity of the constituent units (such as stocks, people, neurons, genes, etc.) of large systems with complicated internal interactions. For this reason, “big time-series data” offer an unprecedented empirical resource for the science of complex systems.

Multiple time series are, in fact, the key ingredient required in order to face one of the main challenges for our modern understanding of real-world complex systems: the identification of an emergent, mesoscopic level of dynamical organization which is intermediate between the microscopic dynamics of individual units (e.g., neurons) and the macroscopic dynamics of the system as a whole (e.g., the brain). Many complex systems are indeed organized in a modular way, with functionally related units being correlated with each other while, at the same time, being relatively less (or even negatively) correlated with functionally dissimilar ones. While the existence of such a modular organization is intuitively plausible, its empirical identification is still an open problem, complicated by the fact that modules are typically emergent, in the sense that they are not evident *a priori* from a local inspection of static, or even dynamic, similarities or connections among individual units. In neuroscience, for instance, “functional

Published by the American Physical Society under the terms of the Creative Commons Attribution 3.0 License. Further distribution of this work must maintain attribution to the author(s) and the published article's title, journal citation, and DOI.

brain networks” are precisely defined by the correlated dynamical activity of neurons, as opposed to “structural brain networks” which are instead defined by static neuronal connections [3]. Remarkably, it has been proposed that the observed divergence between functional and structural brain networks represents a signature of the brain’s many-to-one (degenerate) function-structure relationships which allow diverse functions to arise from a static neuronal anatomy [3]. Similarly, in the analysis of financial markets, it has been observed that groups of correlated stocks evolve in time and only partially overlap with industrial sectors, implying that the (static) industrial classification fails to capture the dynamical modularity of real markets [4–8].

The approaches proposed so far to infer some form of modular or hierarchical organization from multiple time series are based on (necessarily arbitrary) criteria used to filter information [4,5,9]. As we discuss in more detail below, these filtering criteria are either the introduction of thresholds or a geometric embedding in some metric space with predefined properties. Our aim in the present paper is that of going beyond the limitations imposed by these arbitrary criteria. We propose that, both conceptually and algorithmically, the identification of mesoscopic modules, whose dynamical activity is more correlated internally than with that of other modules, requires iterated recursions into many attempted partitions of the system, an inherently nonlocal operation. By their nature, threshold-based or geometric methods are unfortunately not suited to deal with this sort of iterative partitioning problem [10].

Our strategy towards a solution is the adaptation of a different class of rapidly developing techniques, specifically those aimed at identifying the static mesoscopic organization in complex networks, a problem known as “community detection” [11,12]. Communities within networks are groups of nodes that are more densely connected to each other than would be expected under a suitable null hypothesis. Additionally, the nodes within a community are less connected to the nodes within other communities of the same network. Several methods have been proposed over the last decade in order to empirically detect communities within networks. Different techniques have explored different ways to optimize the search over all possible partitions of the system. Conceptually, these methods contain precisely the ingredients that we need in order to solve our problem of identifying the hidden mesoscopic organization encoded within multiple time series. Adapting the existing community detection techniques to deal with time-series data is the main goal of this paper.

While the idea of using community detection algorithms in order to analyze time-series data has already been exploited a few times in the past [13–16], the attempts made so far have basically replaced network data with cross-correlation matrices. Here, we show that this procedure suffers from the limitation that the underlying null hypotheses used in network-based community detection

algorithms are inconsistent with the properties of correlation matrices. We illustrate that one of the undesired consequences is a systematic bias in the search over partitions, which becomes stronger as the heterogeneity of the size of the “true” communities increases.

Here, we propose a solution to this problem by introducing appropriate redefinitions of the so-called modularity [11], the core quantity that most methods aim at maximizing when searching the space of possible partitions. While in ordinary community detection methods the modularity is defined in terms of a null model that is (approximately) correct for networks, in the methods we propose the modularity is defined in terms of different null models that are appropriate for time-series data and therefore dictated by random matrix theory (RMT) [7,17,18]. We also adapt three popular algorithms that have been proposed to find the optimal partition (in networks), i.e., the one that maximizes the modularity. As a result, we end up with three community detection algorithms that are consistent with time-series data and represent the counterparts of the most popular techniques used in network analysis. We also provide extensions to resolve hierarchically nested subcommunities (multiresolution community detection) and “hard” cores versus “soft” peripheries inside communities (multifrequency and time-dependent community detection).

After introducing our theoretical framework, we put special emphasis on financial applications, where the units of the system are assets and the corresponding time series are sequences of logarithmic price increments [4,5,9]. Even though advanced techniques to analyze correlations have been developed in other fields as well, financial time-series analysis is one of the most active domains in this respect (another important example is that of functional brain networks, as we have already mentioned). We show that our methods allow us to efficiently probe the mesoscopic structure of different financial markets and ascertain communities of corporations, based on the time series of their daily stock returns. We uncover a variety of correlations between stocks of different industry sectors, not intuitively obvious from the sectorial taxonomy alone, thus confirming, in a more rigorous manner, the aforementioned result that market correlations only partially overlap with industry classifications. More importantly, the communities we detect after removing noisy and market-wide dependencies turn out to be internally correlated and mutually anticorrelated, a feature of particular relevance for portfolio optimization and risk management. We also analyze the stability of communities over different frequency resolutions and time horizons, thereby identifying groups of “hard stocks” that reside stably in the core of communities and groups of “soft stocks” that alternate between communities.

The rest of the paper is organized as follows: In Sec. II, we briefly describe the most important approaches that have been proposed in order to filter correlation matrices and highlight their issues with characterizing the modular properties of systems described by multiple time series.

In Sec. III, we show that the existing community detection algorithms are based on a null hypothesis that is inconsistent for time-series data, making these methods inadequate as well. In Sec. IV, we then introduce alternative and appropriate null models based on RMT and exploit them in order to redefine three of the most popular community detection algorithms, in a way that makes them consistent with time-series data. In Sec. V, we apply our methods to several time series of daily stock returns, from various financial markets around the globe. In Sec. VI, we analyze the dependence of community structure on the temporal resolution (i.e., the frequency) of the original time series. In Sec. VII, we investigate the evolution of community structure over time. Finally, in Sec. VIII, we summarize our results and provide some conclusions.

II. EXISTING APPROACHES

We start by introducing some useful notation. Let us consider a system with N units. The single time series

$$X_i \equiv \{x_i(1), x_i(2), \dots, x_i(T)\} \quad (1)$$

represents the temporally ordered activity of the i th unit of the system over T time steps. In the case of financial markets, i is typically one particular stock and $x_i(t)$ is the “log-return” of stock i , i.e., the difference between the logarithms of the price of i at times t and $t - 1$ (more details will be given later).

The whole set of N time series, denoted by $\{X_1, X_2, \dots, X_N\}$, describes the synchronous activity of all the units of the system. The vast majority of the available techniques aimed at quantifying the level of mutual dependency within such a set of multiple time series exploit the information encoded in the $N \times N$ cross-correlation matrix. The cross-correlation matrix \mathbf{C} measures the mutual dependencies among N time series on a scale between -1 and 1 . The generic entry of \mathbf{C} is defined as the Pearson correlation coefficient

$$C_{ij} \equiv \text{Corr}[X_i, X_j] \equiv \frac{\text{Cov}[X_i, X_j]}{\sqrt{\text{Var}[X_i] \cdot \text{Var}[X_j]}}, \quad (2)$$

where

$$\text{Cov}[X_i, X_j] \equiv \overline{X_i X_j} - \overline{X_i} \cdot \overline{X_j} \quad (3)$$

is the covariance of X_i and X_j and

$$\text{Var}[X_i] \equiv \sigma_i^2 \equiv \overline{X_i^2} - \overline{X_i}^2 = \text{Cov}[X_i, X_i] \quad (4)$$

is the variance of X_i . In the above equations, the bar denotes a temporal average, i.e.,

$$\overline{X_i} \equiv T^{-1} \sum_{t=1}^T x_i(t), \quad (5)$$

$$\overline{X_i^2} \equiv T^{-1} \sum_{t=1}^T x_i^2(t), \quad (6)$$

$$\overline{X_i X_j} \equiv T^{-1} \sum_{t=1}^T x_i(t) x_j(t). \quad (7)$$

Clearly, the diagonal entries of the correlation matrix are $C_{ii} = 1$.

We will assume, as routinely done in order to filter out the intrinsic heterogeneity of time series, that each series X_i has been standardized by subtracting out the temporal average $\overline{X_i}$ and dividing the result by the standard deviation σ_i , i.e., that X_i has been redefined as $(X_i - \overline{X_i})/\sigma_i$. Then, the following expressions hold:

$$\overline{X_i} = 0, \quad (8)$$

$$\text{Var}[X_i] = \overline{X_i^2} = 1, \quad (9)$$

$$C_{ij} = \text{Cov}[X_i, X_j] = \overline{X_i X_j}. \quad (10)$$

Note that, even though in statistics the notation $\text{Corr}[X_i, X_j]$, $\text{Cov}[X_i, X_j]$ or $\text{Var}[X_i]$ usually denotes a population value, i.e., a theoretical value calculated using the knowledge of the (joint) probability distributions for X_i and X_j , all quantities we have defined so far are instead sample quantities, i.e., measured on the specific realized values of a set of time series. Our choice of a somewhat unconventional notation is merely due to the fact that it allows us to describe various operations more compactly. We will need to denote the population value of a quantity only in a few cases, and when this happens, such population value will coincide with the expected value $\langle f(X, Y, \dots) \rangle$ (over the joint probability distribution of the random variables X, Y, \dots involved) of the corresponding sample quantity $f(X, Y, \dots)$. We will therefore directly express population quantities in terms of expected values when necessary.

We stress that empirical cross-correlation matrices are intrinsically limited by the fact that they assume temporally stationary and linearly interdependent time series. Clearly, both assumptions are, in general, violated in real financial markets and many other complex systems. Nonetheless, cross-correlations are still the most widely used quantity. Improving the definition of correlations is a very important open problem, but it is beyond the scope of this paper. Here, we want to overcome the limitations encountered when the methods introduced so far to process or filter correlation matrices are used in order to identify a mesoscopic modular structure. These current limitations are in place even when correlations are an appropriate measure, i.e., for stationary and linearly interdependent time series. Therefore, our goal is that of introducing a consistent methodology that makes optimal use of correlation matrices in order to resolve the

mesoscopic organization of complex systems. If improved measures of interdependency are introduced, our approach will still represent a valuable guideline in order to implement a consistent community detection framework in that case as well.

In what follows, we review the most important correlation-based approaches and their limitations. We will put special emphasis on financial time series, even if our discussion is more general.

A. Asset graphs

Among the proposed approaches to filter cross-correlation matrices, the simplest one is perhaps that of focusing on the strongest (off-diagonal) correlations by introducing a threshold value and discarding all the correlations below the threshold. The result can be represented as a network, also known as an asset graph (AG) in the econophysics literature [4,19,20], connecting the nodes whose time series are more strongly correlated. Since the method entirely depends on the choice of the threshold, one usually investigates how the properties of the AG change as the threshold is varied. The method is quite robust to noise, precisely because it discards the weakest correlations that are more subject to random fluctuations. However, for the same reason, it fails in detecting a mesoscopic organization (if present) of the system. In fact, the use of a global threshold prevents the identification of modules whose internal correlations, even if below the threshold because they are weak with respect to the strongest ones, are still significantly stronger than the external correlations with different modules. Therefore, while valuable as a filtering technique, the AG discards a significant amount of information and is not best suited to detect emergent groups of correlated time series. We provide additional information about AGs, along with an explicit example, when we analyze real financial data in Sec. V B.

B. Minimal spanning trees

Another filtering approach looks for the minimal spanning tree (MST) obtained again from the strongest correlations, but now retaining only the $N - 1$ correlations that are required for each node to be reachable from any other node via a connected path, while discarding those that produce loops [9]. This procedure automatically produces an agglomerative hierarchical clustering (a dendrogram) of the original time series and requires that the correlation matrix is “renormalized” at each iteration of the clustering according to some protocol (the one having some distinct theoretical advantage is the so-called single-linkage clustering algorithm [9]), until a final filtered matrix is obtained.

The MST method does not require the introduction of an arbitrary threshold, but it assumes that the original correlations are well approximated by the filtered ones. At a geometrical level, this corresponds to the assumption that the metric space in which the original time series are

embedded (via the definition of a proper correlation-based distance) effectively reduces to a so-called ultrametric space where well-separated clusters of points are hierarchically nested within larger well-separated clusters [21].

Even if the method exploits the correlations required for the MST to span the entire set of time series, it discards all the weaker correlations. Moreover, the approximating (renormalized) correlations are progressively more distant from the original ones as higher and higher levels of the taxonomic tree are resolved. This means that the method is more reliable when using the strongest correlations to determine the low-level structure of the taxonomic tree (small clusters of time series), while it is progressively less reliable when using the weaker correlations to determine the high-level taxonomy (medium-sized and large clusters).

With the above warning in mind, the method allows one to identify correlated groups of stocks lying on separate “branches” of the MST or that become disconnected when the associated dendrogram is cut at some level. However, this comes at the price of introducing an arbitrary threshold on the value of the correlation again. Moreover, just like the AG technique, the MST does not compare internal and cross-group correlations (possibly with the aid of a null model) in order to identify emergent mesoscopic modules.

C. Planar maximally filtered graphs

An alternative approach, which is similar in spirit to the MST but discards less information, is the so-called planar maximally filtered graph (PMFG) [22,23]. This method allows one to retain not just the correlations required to form the MST but also a number of additional ones, provided that the resulting structure is a planar graph (a network that can be drawn on a plane without creating intersecting links).

A nice feature of the PMFG is that it always contains the entire MST, so the former provides additional, and not just different, information with respect to the latter. However, this method is also affected by some degree of arbitrariness, which again lies in the properties of the postulated, approximating structure. There is no obvious reason why stocks (or other time series) should find a natural embedding in a bidimensional plane. In fact, the PMFG has also been described as the simplest case of a more general procedure based on the embedding of high-dimensional data in lower-dimensional manifolds with a controllable genus (number of “handles” or “holes”) [23]. The PMFG corresponds to the case when the genus is zero. So the arbitrariness of the method can be rephrased as its dependence on some value of the genus that must be fixed *a priori*.

The method has been extended in a variety of ways in order to produce a nested hierarchy of time series by exploiting the properties of the embedding space [24–26]. However, as with the MST, the target of these methods is that of finding the postulated approximating structure rather

than optimizing the search of groups of time series that are more correlated internally than with each other.

D. Random matrix theory

We finally mention an important technique, based on RMT [7,17,18], which is widely used in order to identify the nonrandom properties of empirical correlation matrices. We use this technique extensively in this paper. A correlation matrix constructed from N completely random time series of duration T has (in the limits $N \rightarrow +\infty$ and $T \rightarrow +\infty$ with $1 < T/N < +\infty$) a very specific distribution of its eigenvalues, known as the Marcenko-Pastur or Sengupta-Mitra distribution [27,28]. This distribution reads

$$\rho(\lambda) = \frac{T}{N} \frac{\sqrt{(\lambda_+ - \lambda)(\lambda - \lambda_-)}}{2\pi\lambda} \quad \text{if } \lambda_+ \leq \lambda \leq \lambda_- \quad (11)$$

and $\rho(\lambda) = 0$ otherwise, where the maximum (λ_+) and minimum (λ_-) eigenvalues are given by

$$\lambda_{\pm} = \left[1 \pm \sqrt{\frac{N}{T}} \right]^2. \quad (12)$$

The bulk of the eigenvalues of an empirical correlation matrix that fall within the range $[\lambda_-, \lambda_+]$ can be considered to be mostly due to random noise. Thus, any eigenvalues larger than the maximum eigenvalue λ_+ predicted by the Marcenko-Pastur distribution are deemed to represent meaningful structure in the data [6–8]. That being the case, any empirical correlation matrix \mathbf{C} can be decomposed as the sum of two matrices:

$$\mathbf{C} = \mathbf{C}^{(r)} + \mathbf{C}^{(s)}, \quad (13)$$

where (using $\langle \text{bra} |$ and $| \text{ket} \rangle$ notation)

$$\mathbf{C}^{(r)} \equiv \sum_{i: \lambda_i \leq \lambda_+} \lambda_i |v_i\rangle \langle v_i| \quad (14)$$

is the “random” component constituted from the eigenvalues $\{\lambda_i\}$ less than or equal to λ_+ (usually, the eigenvalues smaller than λ_- are included as well) and their corresponding eigenvectors $\{|v_i\rangle\}$, and $\mathbf{C}^{(s)} = \mathbf{C} - \mathbf{C}^{(r)}$ is the “structured” component constituted from the remaining eigenvalues corresponding to eigenvalues larger than λ_+ .

The deviation of the spectra of real correlation matrices from the RMT prediction provides an effective way to filter out noise from empirical data, and it also illustrates some robust property of financial markets. For instance, in Fig. 1, we superimpose the eigenvalue density of the empirical correlation matrix obtained from $T = 2500$ log-returns of daily closing prices of $N = 445$ stocks of the S&P 500 index (from 2001 to 2011) and the corresponding expectation given by the Marcenko-Pastur distribution with the same values of N and T . As also observed in a multitude of previous studies [4,5], a typical feature of the spectrum of

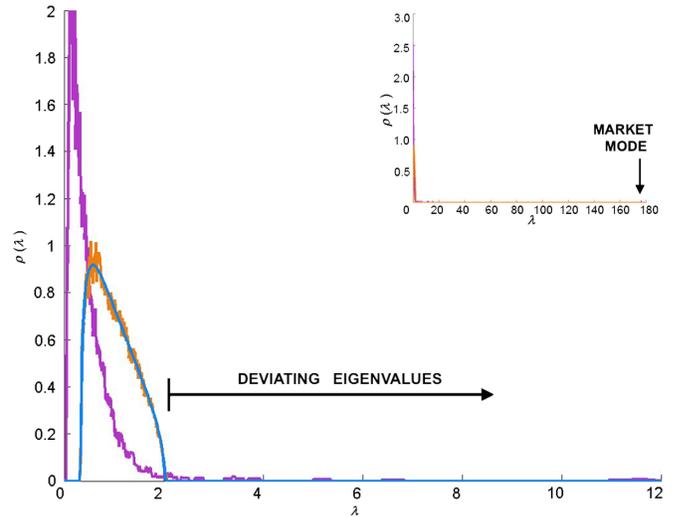


FIG. 1. The eigenvalue density of the empirical correlation matrix of $T = 2500$ log-returns (of daily closing prices from 2001Q4 to 2011Q3) for $N = 445$ stocks of the S&P 500 index (purple) and the Marcenko-Pastur prediction for a random correlation matrix with the same values of N and T (blue), denoting a maximum expected eigenvalue of approximately 2. The orange plot is the eigenvalue density obtained by randomly shuffling (i.e., permuting with uniform probability) the empirical increments within each of the observed time series, confirming the agreement with random matrix theory for uncorrelated data. The inset is the fully zoomed-out version of the plot, showing that the empirical correlation matrix has a maximum eigenvalue of about 175 (“market mode”), as well as a handful of other leading eigenvalues above the predicted maximum value.

empirical correlation matrices is that the largest observed eigenvalue λ_m is much larger than all other eigenvalues (see inset of Fig. 1). The corresponding eigenvector $|v_m\rangle$ has all positive signs, and one can therefore identify this eigen-component of the correlations as the so-called market mode [4,5], i.e., a common factor influencing all stocks within a given market. With this interpretation, the bulk of the correlation between pairs of stocks is attributed to a single common factor, much as all boats in a harbor will rise and fall with the tide.

In order to clearly see which “boats” are rising and falling relative to one another, one must subtract out the common “tide,” which in terms of the correlation matrix leads to the further decomposition

$$\mathbf{C} = \mathbf{C}^{(r)} + \mathbf{C}^{(g)} + \mathbf{C}^{(m)}, \quad (15)$$

where we have rewritten the structured component as $\mathbf{C}^{(s)} = \mathbf{C}^{(g)} + \mathbf{C}^{(m)}$, with

$$\mathbf{C}^{(m)} \equiv \lambda_m |v_m\rangle \langle v_m| \quad (16)$$

(representing the market mode) and

$$\mathbf{C}^{(g)} \equiv \sum_{i: \lambda_+ < \lambda_i < \lambda_m} \lambda_i |v_i\rangle \langle v_i| \quad (17)$$

(representing the remaining correlations).

The correlations embodied by $\mathbf{C}^{(g)}$ act neither at the level of individual stocks (uncorrelated noise) nor at that of the entire market. Such correlations act at the level of subgroups of stocks within a market, and they are often referred to as the “group” mode [4,28]. The eigenvectors contributing to $\mathbf{C}^{(g)}$ have alternating signs, and this allows the identification of groups of stocks that are influenced in a similar manner by one or more common factors [6–8]. Broadly speaking, these groups are expected to reflect some sectorial or subsectorial classification of stocks according to their industrial category; however, the overlap between nominal asset classes and groups of empirically correlated stocks is only partial [6–8].

We should, at this point, stress that the above discussion makes some strong assumptions, which have been recently criticized. In particular, the interpretation of the largest eigenvalue in terms of a market mode and the assumption that the elimination of the market and noise modes does not alter the information present in the remaining subspace are not correct, in general, and are sometimes only approximate [29,30]. Moreover, the eigencomponents of the correlation matrix, and consequently the filtered correlation matrix itself, can end up not being proper correlation matrices, and alternative constructions enforcing the required properties have been proposed [31–34]. Finally, the way to filter out the market and noise modes is not unique [35].

Bearing these limitations in mind, RMT is still to be considered a valuable tool to filter empirical correlation matrices and clean them from both stock-level (random) and market-wide fluctuations. However, after this preprocessing, filtered correlation data still need to be analyzed according to the particular research question. For instance, the matrix $\mathbf{C}^{(g)}$ is often processed further and used as an alternative, filtered input in all the algorithms (AG, MST, and PMFG) described above. So RMT alone is not enough in order to resolve the mesoscopic organization of markets, in the sense defined above.

III. COMMUNITY DETECTION IN GRAPHS AND ITS INCONSISTENCY WITH CORRELATION MATRICES

In the previous section, we clarified that many of the available techniques used to identify the most relevant correlations are not designed to isolate groups of time series whose dynamical activity is more correlated internally than with that of other groups. At an abstract level, achieving this task would require iterated recursions into many attempted partitions of the system, an inherently nonlocal and computationally demanding operation. Notably, an entire branch of network science is devoted to an analogous problem, known as community detection [11]. In this

section, we briefly illustrate the principles of community detection in networks and show how that knowledge can, in principle, be transferred to our initial problem, namely, the identification of a mesoscopic organization across multiple time series. We also show that despite the many progressive inroads made in this direction so far, they often rely on an inherently biased approach.

A. Community detection in networks

In network analysis, community detection is the process of identifying relatively dense clusters of nodes. There has been a flurry of research in the area of community detection over the last decade [11]. In this paper, we focus on the method of modularity optimization [36], which is one of the most popular methods identifying nonoverlapping communities. It should be noted that various alternative methods other than modularity optimization exist, including techniques that resolve overlapping communities [11,37]. However, this method has the advantage of being based on a null model, acting as a community-free benchmark to which the real network is compared. It is the appropriate modification of such benchmarks that will lead us, in Sec. IV, to a redefinition of modularity optimization methods valid for correlation matrices.

We restrict ourselves to undirected networks, since they exhibit the same symmetry property as correlation matrices. Given a network with N nodes, one can introduce a number of partitions of the N nodes into nonoverlapping sets. Each such partition can be mathematically represented by an N -dimensional vector $\vec{\sigma}$, where the i th component σ_i denotes the set in which node i is placed by that particular partition. Then, one can introduce the so-called modularity $Q(\vec{\sigma})$ as a measure of the effectiveness of a particular partition $\vec{\sigma}$ in identifying densely connected groups of nodes. The process of modularity optimization seeks to find the optimal partition that maximizes the value of $Q(\vec{\sigma})$, by varying the communities to which the different nodes of the network belong. The modularity $Q(\vec{\sigma})$ is expressed in the form

$$Q(\vec{\sigma}) = \frac{1}{A_{\text{tot}}} \sum_{i,j} [A_{ij} - \langle A_{ij} \rangle] \delta(\sigma_i, \sigma_j), \quad (18)$$

where, here and throughout the paper, the sum is intended to run over *all* pairs of nodes even if we are considering undirected networks, and we are also including the diagonal elements corresponding to $i = j$ since many expressions become simpler with this choice. The meaning of the different terms of the above expression is as follows. The delta function is $\delta(\sigma_i, \sigma_j) = 1$ if $\sigma_i = \sigma_j$ and $\delta(\sigma_i, \sigma_j) = 0$ if $\sigma_i \neq \sigma_j$, ensuring that only nodes within the same community contribute to the sum. For binary networks, A_{ij} is the entry of the adjacency matrix representing the presence ($A_{ij} = 1$) or absence ($A_{ij} = 0$) of a link between

nodes i and j in the observed network. The initial prefactor works to normalize the value of $Q(\vec{\sigma})$ between -1 and 1 , where $A_{\text{tot}} \equiv \sum_{i,j} A_{ij} = 2m$ is twice the total number m of links.

The term $\langle A_{ij} \rangle$ is a key element determining the outcome of the entire community detection process. It mathematically represents a null model for the network, i.e., an expectation for A_{ij} under some suitable null hypothesis. The most popular null model for a binary network, known as the configuration model, is one where the expected value $\langle k_i \rangle$ of the degree k_i (number of links) of each node i is equal to the value $\sum_{j=1}^N A_{ij}$ observed in the real network and where the topology is otherwise completely random. This null hypothesis ensures that the local heterogeneity of nodes, e.g., the fact that more popular people naturally have more friends in social networks, is appropriately accounted for. Mathematically, this model is approximately (i.e., only when the heterogeneity of the degrees is weak [38,39]) represented by the expression

$$\langle A_{ij} \rangle = \frac{k_i k_j}{2m}, \quad (19)$$

which gives a rough estimate of the probability that nodes i and j are connected, under the null hypothesis that the observed network's structure is completely explained on the basis of the different degrees of vertices.

For weighted networks, A_{ij} denotes the weight of the link between nodes i and j , k_i is called the *strength* of node i , and $2m$ is twice the total weight (of all links in the network). Still, Eq. (19) is used without modifications [11] to determine the (again approximate [39,40]) expected edge weight under the null hypothesis that the network's structure is completely explained on the basis of the observed strengths of all vertices.

The accuracy and usefulness of the results obtained from the process of modularity optimization depend heavily on the choice and suitability of the null model. When the null hypothesis is true, no higher-order patterns (including communities) are present. Consistently, one expects the modularity in Eq. (18) to be close to zero for every partition. In maximizing the modularity for a network that does have community structure, the nodes that are more tightly connected than one would expect on the basis of their individual characteristics will be clustered together in the same community, while the nodes for which the opposite occurs will be placed in different communities.

It should be noted that, in the context of network analysis, the modularity function defined in Eq. (18) suffers from a main drawback: It cannot resolve communities below a typical scale [41]. This resolution limit was proven to be rooted in the specific mathematical form of Eq. (19) used to represent the null model. However, it was not proven to be due to the concept of the null model itself, i.e., to the choice of comparing the real network with an

ensemble of graphs with given degrees (or strengths). In particular, we stress again that Eq. (19) only approximately represents such an ensemble, the exact formula being a more complicated nonlinear equation [38–40]. Whether the resolution limit disappears if the exact expression is used in place of Eq. (19) has never been investigated. Rather, it has been proposed [42] that a way to change the resolution of the community detection is the introduction of an extra resolution parameter $\phi > 0$ in the null model, i.e., replacing Eq. (19) with

$$\langle A_{ij} \rangle = \phi \frac{k_i k_j}{2m}. \quad (20)$$

Many studies have indeed shown that, as ϕ is varied, different hierarchical levels of the community structure can be revealed, so a so-called multiresolution method can be obtained [42–44]. In general, multiresolution methods can resolve smaller subcommunities, which are nested inside larger communities. One should, however, bear in mind that the resolution parameter was originally introduced in an *ad hoc* fashion and without a theoretical foundation, its main justification being an agreement *a posteriori* with the hierarchical community structure expected in some real-world networks. Only later was it shown to have some physical interpretation in terms of an inverse time required to explore the network under certain assumptions [44]. When extending modularity-based algorithms to the analysis of multiple time series, we will address the problem of multiresolution community detection in a fundamentally different way, which avoids *ad hoc* parameters and is theoretically consistent with the properties of correlation matrices (see Sec. IV C).

B. Inconsistency of modularity for cross-correlation matrices

The appealing properties of community detection in networks clearly have the potential to solve our initial problem of finding groups of time series that are more correlated than we would expect. However, one should be very careful in identifying the correlation-based problem with the network-based one. A naive approach would be that of treating the empirical correlation matrix \mathbf{C} as a weighted network and looking for communities using the modularity as defined in Eq. (18), i.e., setting $A_{ij} = C_{ij}$. This would result in a modularity of the form

$$Q(\vec{\sigma}) = \frac{1}{C_{\text{norm}}} \sum_{i,j} [C_{ij} - \langle C_{ij} \rangle] \delta(\sigma_i, \sigma_j), \quad (21)$$

where $C_{\text{norm}} = \sum_{i,j} C_{ij}$, $\langle C_{ij} \rangle = k_i k_j / C_{\text{norm}}$ and $k_i = \sum_{j=1}^N C_{ij}$. This idea has been recently exploited, sometimes with modifications, to study communities of interest rates [14] and stocks [13,15,16] in financial markets.

Unfortunately, although the above approach has made a lot of headway, it suffers from some fundamental flaws and can lead to biased results, as we now show. The problem arises because the null model defined in Eq. (19), while (approximately [38–40]) valid when the matrix \mathbf{A} describes a network, is inconsistent if \mathbf{A} is replaced by a correlation matrix \mathbf{C} . Indeed, note that if $A_{ij} = C_{ij}$ and if X_i denotes a standardized time series i (see Sec. II), then Eq. (10) implies

$$k_i \equiv \sum_{j=1}^N C_{ij} = \sum_{j=1}^N \text{Cov}[X_i, X_j] = \text{Cov}[X_i, X_{\text{tot}}], \quad (22)$$

where $X_{\text{tot}} = \{x_{\text{tot}}(1), x_{\text{tot}}(2), \dots, x_{\text{tot}}(T)\}$ is the time series of the total increment $x_{\text{tot}}(t) \equiv \sum_{j=1}^N x_j(t)$. Note that, even if all X_i 's are standardized, X_{tot} has zero mean but nonunit variance and is therefore *not* standardized. Similarly,

$$2m = \sum_{i=1}^N k_i = \text{Cov}[X_{\text{tot}}, X_{\text{tot}}] = \text{Var}[X_{\text{tot}}]. \quad (23)$$

It then follows that

$$\begin{aligned} \frac{k_i k_j}{2m} &= \frac{\text{Cov}[X_i, X_{\text{tot}}] \cdot \text{Cov}[X_j, X_{\text{tot}}]}{\text{Var}[X_{\text{tot}}]} \\ &= \text{Corr}[X_i, X_{\text{tot}}] \cdot \text{Corr}[X_j, X_{\text{tot}}]. \end{aligned} \quad (24)$$

We therefore arrive at an important conclusion: For correlation matrices, the “naïve” modularity, as ordinarily defined in Eq. (18) with the ordinary specification given in Eq. (19), corresponds to the following null hypothesis:

$$\langle C_{ij} \rangle_{\text{naive}} = \text{Corr}[X_i, X_{\text{tot}}] \cdot \text{Corr}[X_j, X_{\text{tot}}]. \quad (25)$$

When used within the modularity function, the above null model will not necessarily give more importance to pairs of strongly correlated time series but rather to pairs of time series whose “direct” correlation C_{ij} is larger than the product of the correlations of each time series with the “common signal” X_{tot} . On the other hand, if we want to detect communities of time series that are empirically more correlated than expected under the hypothesis that all time series are independent of each other, we know that the correct null model (at least for infinitely long time series, a hypothesis that we will relax later) is

$$\langle C_{ij} \rangle = \delta_{ij}; \quad (26)$$

i.e., the expected correlation matrix $\langle \mathbf{C} \rangle$ should be the $N \times N$ identity matrix \mathbf{I} . Other acceptable forms of $\langle C_{ij} \rangle$ based on realistic properties of correlation matrices will be discussed later.

The origin of the problematic discrepancy between Eqs. (25) and (26) is the fact that the null model defined in Eq. (19) is meant to represent networks with given degrees, i.e., matrices with given column and row sums. Any matrix that matches this constraint is admissible, in the sense that it represents a possible [45] network consistent with the hypothesis that degrees are an important structural constraint. By contrast, sums over rows or columns of correlation matrices do not represent any meaningful constraint, as evident from Eq. (22). Moreover, not every symmetric real matrix with given row and column sums is a possible correlation matrix. Correlation matrices must also be positive-semidefinite, i.e., have non-negative eigenvalues. A little algebra shows that Eq. (25) fulfills this property but in a very extreme way: The eigenvalues of the matrix having elements $\langle C_{ij} \rangle_{\text{naive}}$ are $\lambda = 0$ (with multiplicity $N - 1$) and

$$\lambda = \sum_{i=1}^N (\text{Corr}[X_i, X_{\text{tot}}])^2 \quad (27)$$

(with multiplicity 1). This result holds irrespective of the original data, e.g., also for correlated and finite-length time series. Our discussion of the spectrum of realistic correlation matrices in Sec. IID strongly indicates that a sensible null model for correlation matrices should feature an eigenvalue distribution that is not easily reducible to the extremely simple one found above.

Similar conceptual limitations are also encountered in more sophisticated null models, which while allowing for both positive and negative link weights [46], still consider all possible matrices (many of which are inconsistent with correlation matrices) with given sums over rows and columns. More importantly, the above problems cannot be solved by the introduction of resolution parameters. If, in analogy with Eq. (20), we consider the generalized null model

$$\langle C_{ij} \rangle_{\text{naive}} = \phi \cdot \text{Corr}[X_i, X_{\text{tot}}] \cdot \text{Corr}[X_j, X_{\text{tot}}] \quad (28)$$

(with $\phi > 0$), we are still left with an expression that cannot be reduced to Eq. (26) or some other meaningful alternatives, which we will introduce later in Sec. IVA. For instance, the eigenvalues become $\phi\lambda$, where λ still takes only the two values shown above. Further aspects of this limitation are explicitly illustrated in a benchmark case below, and they imply that appropriate multiresolution community detection methods for correlation matrices should be implemented in a completely different way (see Sec. IV C).

C. Bias produced by the naive approach

To have an idea of the consequence of using the naive approach, i.e., the application of a network-based modularity directly to a cross-correlation matrix, we consider an

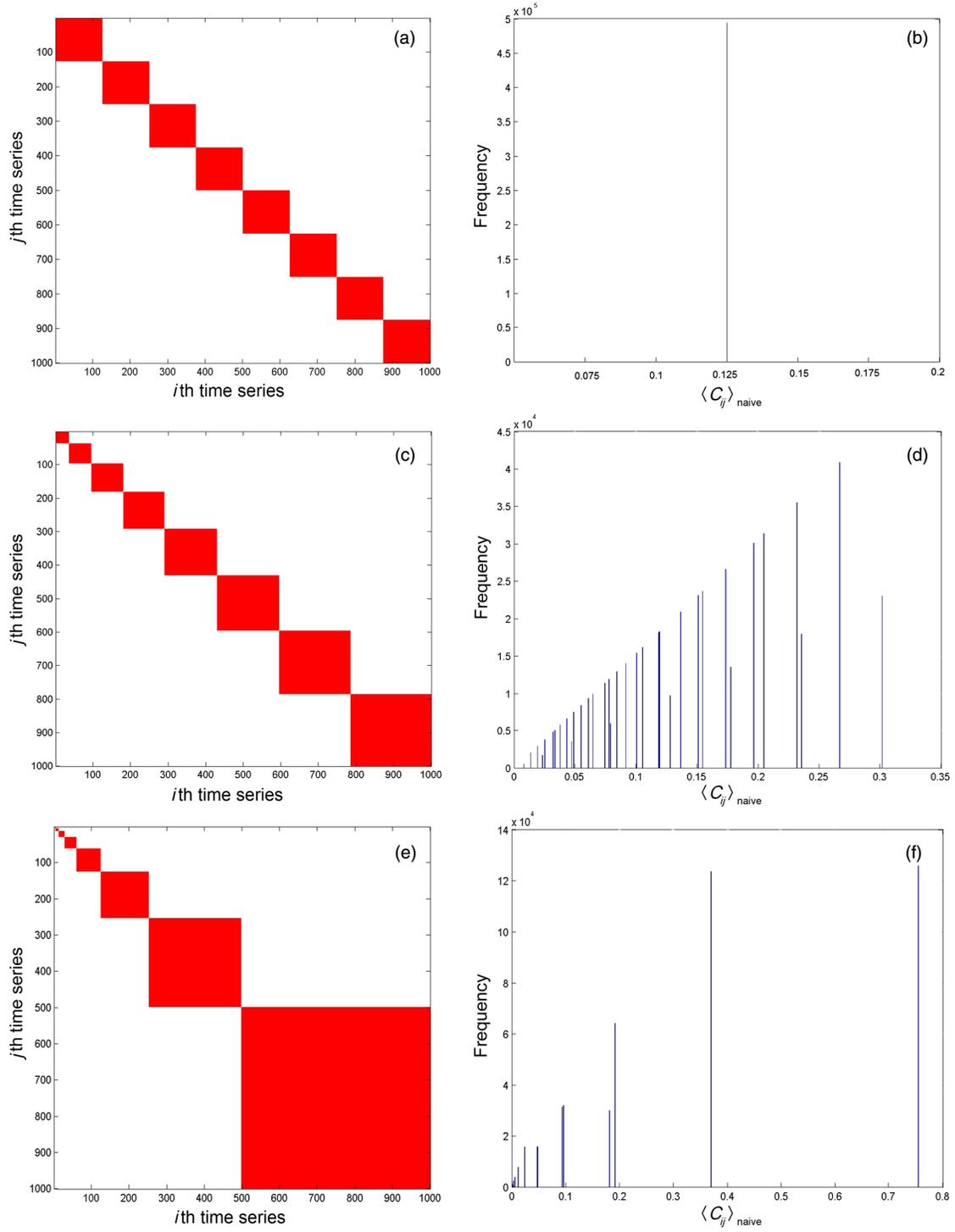


FIG. 2. The biasing effect of the heterogeneity of community size on the naive (network-based) modularity. The left panels show the entries $\delta(\sigma_i^*, \sigma_j^*)$ of three benchmark community matrices (white = 0, red = 1), each with $N = 1000$ time series and $c = 8$ communities of increasingly heterogeneous sizes. The right panels show the corresponding distribution of the off-diagonal entries $\langle C_{ij} \rangle_{\text{naive}}$ of the null model defined in Eq. (32), with $\phi = 1$ (the bin size of the histograms is of the order of 10^{-5} , which makes the distributions correctly normalized). For perfectly homogeneous community sizes, i.e., if each community contains exactly 125 time series (a), the distribution has a single peak at $1/c = 0.125$ (b). For moderately heterogeneous sizes, i.e., if the eight communities contain 35, 60, 85, 110, 140, 165, 190, and 215 time series, respectively (c), the distribution has several peaks (d) coming from the 64 different combinations of $n_{\sigma_i^*} n_{\sigma_j^*}$ in Eq. (32). For strongly heterogeneous sizes, i.e., if the eight communities contain 4, 8, 16, 32, 64, 128, 246, and 502 time series, respectively (e), the distribution still has 64 different peaks but is much broader (f). The two dominant peaks are located at $\langle C_{ij} \rangle_{\text{naive}} = 0.7536$ (coming from pairs of time series inside the largest community) and $\langle C_{ij} \rangle_{\text{naive}} = 0.3694$ (coming from pairs of time series across the largest and the second-largest communities).

ideal benchmark case where N infinitely long time series are divided into c “true” communities, specified by a true partition $\vec{\sigma}^*$. We assume that each community A is made of n_A standardized time series (with $\sum_{A=1}^c n_A = N$) that are perfectly correlated with each other and completely uncorrelated to the time series in other communities, i.e.,

$$C_{ij} = \text{Corr}[X_i, X_j] = \text{Cov}[X_i, X_j] = \delta(\sigma_i^*, \sigma_j^*). \quad (29)$$

In such a case,

$$\text{Cov}[X_i, X_{\text{tot}}] = \sum_{j=1}^N \text{Cov}[X_i, X_j] = n_{\sigma_i^*} \quad (30)$$

(where $n_{\sigma_i^*}$ is the number of time series in the community of the time series i) and

$$\text{Var}[X_{\text{tot}}] = \sum_{i,j} \text{Cov}[X_i, X_j] = \sum_{A=1}^c n_A^2. \quad (31)$$

From the last two equations, it follows that Eq. (25), or more generally Eq. (28), can be rewritten as

$$\langle C_{ij} \rangle_{\text{naive}} = \phi \frac{n_{\sigma_i^*} n_{\sigma_j^*}}{\sum_{A=1}^c n_A^2} \quad (32)$$

(with $\phi > 0$), which is the fundamental result showing the inconsistency of the naive approach and the nature of the resulting bias. Equation (32) can never lead to the correct expectation (26) because it cannot produce off-diagonal zeros. If there are c equally sized communities of $n = N/c$ time series each, then $\langle C_{ij} \rangle_{\text{naive}} = \phi/c$ for all i, j , i.e., the distribution of $\langle C_{ij} \rangle_{\text{naive}}$ has a single peak and zero standard deviation. In this case, apart from the minor [47] problem of nonunit diagonal entries, the use of $\langle C_{ij} \rangle_{\text{naive}}$ in Eq. (18) can still be justified on the basis of the fact that ϕ/c is a constant term having no effect on the modularity maximization. However, for heterogeneously sized communities, Eq. (32) does not lead to a mere overall shift in the modularity. As the size heterogeneity increases, the distribution of the off-diagonal entries of $\langle C_{ij} \rangle_{\text{naive}}$ will become broader. In general, $\langle C_{ij} \rangle_{\text{naive}}$ is larger for pairs of time series belonging to larger communities. This effect is shown in Fig. 2 for three choices of benchmark communities.

The above consideration implies that the standard deviation (irrespective of the average) of the off-diagonal ($i \neq j$) entries of $\langle C_{ij} \rangle_{\text{naive}}$ can be taken as a quantitative measure of the bias induced by Eq. (32). This definition depends linearly on the multiresolution parameter ϕ . Alternatively, the coefficient of variation (standard deviation divided by average value) of the off-diagonal entries of $\langle C_{ij} \rangle_{\text{naive}}$ is a measure of the relative bias of the naive approach and is independent of ϕ . One should bear in mind that when the value of the coefficient of variation is

much lower than 1, the heterogeneity is moderate, while when it approaches or exceeds 1, then the heterogeneity is such that the average value is no longer representative of the distribution.

In Fig. 3, we show both the bias (for $\phi = 1$) and the relative bias as a function of size heterogeneity, the latter being, in turn, defined as the coefficient of variation of community size. We see that the (relative) bias first steadily increases as the size heterogeneity increases from zero to approximately 2 and then decreases when the heterogeneity further increases. This decrease corresponds to entering an extremely heterogeneous regime where there is a giant community of $O(N)$ nodes and other very small communities of $O(1)$ nodes. In this regime, the effective number of communities is practically 1, and the distribution of $\langle C_{ij} \rangle_{\text{naive}}$ becomes sharp again, as most entries have the same value. So, for a very broad range of heterogeneity (say, when the coefficient of variation of community sizes is between 0.5 and 2.5), the (relative) bias is very strong. In this regime, Eq. (32) gives a prediction $\langle C_{ij} \rangle_{\text{naive}} \approx 0$ (close to the correct expectation) only for pairs of time series belonging to the smallest community. For such time series, the difference $C_{ij} - \langle C_{ij} \rangle_{\text{naive}}$ is still close to 1, and one therefore expects that the smallest community will be detected correctly. However, for time series belonging to larger communities, $\langle C_{ij} \rangle_{\text{naive}}$ increases, progressively biasing the community detection. For the largest community, the expected internal correlation is always larger than the correlation among any pair of communities, so $C_{ij} - \langle C_{ij} \rangle_{\text{naive}}$ is very low and this community is paradoxically difficult to detect.

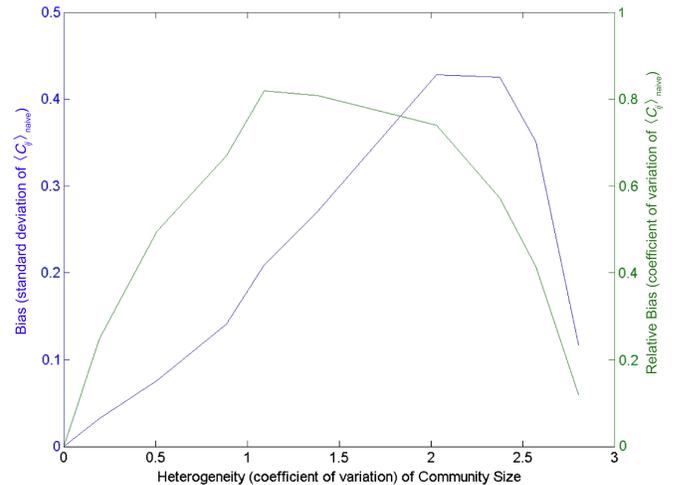


FIG. 3. Dependence of the (relative) bias of the naive approach on the heterogeneity (coefficient of variation) of community size, for various benchmarks with $N = 1000$ time series and $c = 8$ communities. The bias is defined as the standard deviation (coefficient of variation) of the off-diagonal entries $\langle C_{ij} \rangle_{\text{naive}}$ of the null model defined in Eq. (32) with $\phi = 1$, while the relative bias is defined as the coefficient of variation of the same entries (and is independent of ϕ).

It should be noted that the use of the multiresolution parameter ϕ does not help reduce the relative bias, as the latter is independent of ϕ . In order to reduce the absolute bias (which for $\phi = 1$ has values around 0.3 in the relevant regime; see Fig. 3) to small values (say, of the order of 0.01), ϕ should be set to very small values (around 0.03), which is another way of saying that the null model in Eq. (28) should effectively be replaced by that in Eq. (26) (we recall that we are referring only to the off-diagonal entries here), thus confirming our previous discussion.

The above results lead us to conclude that the ordinary definition of modularity, even with the introduction of a multiresolution parameter, cannot properly detect communities. This limitation would systematically bias any modularity-based community detection algorithm. It is therefore clear that ordinary network-based clustering methods, when used with correlation matrices, lead to incorrect results. In the rest of the paper, we try to overcome this limitation.

IV. REDEFINING COMMUNITY DETECTION METHODS FOR MULTIPLE TIME SERIES

We now come to our most important results, i.e., the introduction of improved and consistent methods to cluster multiple time series using appropriate null models. In Sec. IVA we give three redefinitions of the modularity $Q(\vec{\sigma})$ that make use of the results of RMT, which we summarized in Sec. IID. In Sec. IVB, we introduce the correlation-based counterparts of three of the most popular community detection algorithms used in network analysis. In Sec. IVC, we discuss how these algorithms can be further extended in order to obtain appropriate, multiresolution community detection methods. Finally, in Sec. IVD, we benchmark our methods on various test cases.

A code implementing all the methods discussed in this section (with the adaptations discussed in the Appendix) is available here [48,49].

A. Correlation-based redefinitions of modularity

From our previous discussion, it should be clear that simply replacing network data with correlation matrices in Eq. (18) leads to Eq. (21) where C_{norm} is $\sum_{i,j} C_{ij}$ and the null model $\langle C_{ij} \rangle$ is incorrectly given by Eq. (25). We now introduce three redefinitions of modularity based on appropriate null models. The end result of this redefinition will be a set of modularity functions that correctly identify communities of correlated time series. For compactness, we postpone the possible (re)definition of C_{norm} to the end of this discussion, in Sec. IVA 4.

1. Infinite time series without global mode

We have already noted that, for infinitely long time series, the correct expression corresponding to the null hypothesis of independency is given by Eq. (26). This leads us to a first redefinition of modularity with expectation $\langle C_{ij} \rangle_1 \equiv \delta_{ij}$, i.e.,

$$\begin{aligned} Q_1(\vec{\sigma}) &= \frac{1}{C_{\text{norm}}} \sum_{i,j} [C_{ij} - \delta_{ij}] \delta(\sigma_i, \sigma_j) \\ &= \frac{1}{C_{\text{norm}}} \sum_{i,j} C_{ij}^{(\delta)} \delta(\sigma_i, \sigma_j), \end{aligned} \quad (33)$$

where $\mathbf{C}^{(\delta)} \equiv \mathbf{C} - \mathbf{I}$ (\mathbf{I} being the $N \times N$ identity matrix), so $C_{ii}^{(\delta)} = 0$.

2. Finite time series without global mode

For finite-length independent time series, we should further modify our null model to one which anticipates a certain amount of noise, as determined by RMT (see Sec. IID). In such a case, we know that the correct null hypothesis is $\langle C_{ij} \rangle_2 \equiv C_{ij}^{(r)}$, where $\mathbf{C}^{(r)}$ is given by Eq. (14). This gives us a second redefinition of modularity for dealing with noisy correlation matrices:

$$\begin{aligned} Q_2(\vec{\sigma}) &= \frac{1}{C_{\text{norm}}} \sum_{i,j} [C_{ij} - C_{ij}^{(r)}] \delta(\sigma_i, \sigma_j) \\ &= \frac{1}{C_{\text{norm}}} \sum_{i,j} C_{ij}^{(s)} \delta(\sigma_i, \sigma_j). \end{aligned} \quad (34)$$

Note that now, in general, $C_{ii}^{(s)} \neq 0$ as a result of the eigendecomposition defined in Eq. (13). However, the diagonal terms with $i = j$ give an irrelevant constant contribution to the modularity due to the fact that $\delta(\sigma_i, \sigma_i) = 1$ for all i , independently of the particular partition $\vec{\sigma}$. This makes the above definition well defined even in the presence of nonzero diagonal entries.

3. Finite time series with global mode

Lastly, we consider the case where we expect an overall level of positive correlation among all time series, or “global mode.” For instance, we have already mentioned that in financial markets, the presence of the market mode (see Sec. IID) generally results in a positive correlation affecting all pairs of stocks altogether. The corresponding dominant positive component $\mathbf{C}^{(m)}$ of \mathbf{C} would make $Q(\vec{\sigma})$ maximized by the (trivial) partition where all time series are in the same community. In order to detect nontrivial communities, we can choose a null model that includes both the random component of the correlation matrix and the global or market mode, i.e., $\langle C_{ij} \rangle_3 \equiv C_{ij}^{(r)} + C_{ij}^{(m)}$, where $\mathbf{C}^{(r)}$ and $\mathbf{C}^{(m)}$ are given by Eqs. (14) and (16), respectively. This yields our third and final formulation for the modularity:

$$\begin{aligned} Q_3(\vec{\sigma}) &= \frac{1}{C_{\text{norm}}} \sum_{i,j} [C_{ij} - C_{ij}^{(r)} - C_{ij}^{(m)}] \delta(\sigma_i, \sigma_j) \\ &= \frac{1}{C_{\text{norm}}} \sum_{i,j} C_{ij}^{(g)} \delta(\sigma_i, \sigma_j). \end{aligned} \quad (35)$$

In this case as well, $C_{ii}^{(g)} \neq 0$ as a result of the eigendecomposition defined in Eq. (15), but this does not affect the outcome of the community detection.

The above definition is now explicitly aimed at detecting mesoscopic communities, which are in between the “microscopic” level of unit-specific noise and the “macroscopic” level of system-wide fluctuations. While the existence of the market mode is well established in finance, for other types of time series it might be inappropriate to postulate the existence of a global mode. However, we also expect that, whenever the use of $Q_1(\vec{\sigma})$ or $Q_2(\vec{\sigma})$ yields only a single community, the most plausible reason is the existence of a global mode. Accordingly, we expect that the use of $Q_3(\vec{\sigma})$ might be the most appropriate way to filter out global dependencies for a variety of systems, not only for financial markets. Moreover, as we discuss at length in Sec. IV C, iteratively filtering out the global mode from the correlation matrices restricted to individual communities can result in the definition of a useful multiresolution method to resolve multiple hierarchical levels of community structure, if present.

4. A unified redefinition

For simplicity, in what follows, it is useful to express the three definitions of modularity we gave in Eqs. (33), (34), and (35) in unified form:

$$Q_l(\vec{\sigma}) \equiv \frac{1}{C_{\text{norm}}} \sum_{i,j} C_{ij}^{(l)} \delta(\sigma_i, \sigma_j), \quad (36)$$

where

$$\mathbf{C}^{(l)} \equiv \mathbf{C} - \langle \mathbf{C} \rangle_l = \begin{cases} \mathbf{C}^{(\delta)} & l = 1 \\ \mathbf{C}^{(s)} & l = 2 \\ \mathbf{C}^{(g)} & l = 3. \end{cases} \quad (37)$$

In what follows, given a choice of l , we will refer to $\mathbf{C}^{(l)}$ as the “filtered” correlation matrix.

The overall constant C_{norm} has no role in determining the final partition, but it does have a role when different systems, or different snapshots of the same system (including dynamical analyses of community structure), are compared. For simplicity, we keep the same definition as in Eq. (21), i.e.,

$$C_{\text{norm}} \equiv \sum_{i,j} C_{ij} = \text{Var}[X_{\text{tot}}]. \quad (38)$$

This definition implies that the modularity is the sum of intracommunity (filtered) correlations, divided by the variance of the total increment X_{tot} . This variance is a natural measure of the volatility of the system over the considered time window, which in the case of financial time series, is an important property of the market. In other words, Eq. (38) automatically controls for the volatility of the system, a feature that is typically desirable when

analyzing the evolution of (the community structure of) wildly fluctuating systems. However, in some cases it might be interesting to compare the above modularity with one calculated using a different definition of C_{norm} , e.g., one that does not control for the volatility.

It should be noted that the above definition is such that the typical (for real-world systems like financial markets) values of the modularity defined in Eq. (36) will tend to be much lower than the typical (for real-world networks) values of the modularity defined in Eq. (18), even for systems with well-defined communities. One should bear this consideration in mind when interpreting the (maximized) modularity value as a measure of the strength of community structure in the system. Unlike its network counterpart, our definition of the modularity does not quantify the strength of community structure in an absolute scale between -1 and $+1$. It only has a meaning in relative terms, and the more information is contained in the null model, the lower the value of the resulting modularity.

We remind the reader of the fact that, since the results of RMT used in the above definition hold only in the regime where N and T are both large (with $T > N$), we require the original time series to respect these conditions. The requirement $T > N$ is sometimes referred to as the “curse of dimensionality” in the literature since it implies that, in order to study the cross-correlations of a large set of time series, one needs to extend the time interval so much that the assumption of stationarity (implicit, as we mentioned, in the definition of cross-correlations themselves) is violated. On the other hand, choosing sufficiently short time intervals to make the time series approximately stationary implies that the number N of time series is severely reduced. One should therefore choose the data in such a way that a reasonable compromise is achieved. This is an ordinary trade-off to be made in the analysis of any empirical (financial) cross-correlation matrix.

We finally stress that the three RMT-based null models we have adopted do not represent the only possible choices. One might, for instance, exploit more sophisticated results [29–35] and introduce refined null models that overcome some of the limitations of RMT that we mentioned in Sec. II D. These alternative choices can then be incorporated into our approach by redefining $\langle \mathbf{C} \rangle_l$ and, consequently, $\mathbf{C}^{(l)}$. Exploring the entire space of possibilities is beyond the scope of this paper. The key point we are stressing here is that, whatever the choice of the null model, it must respect some realistic properties of correlation matrices. The network-based definition of modularity, which has been used so far, does not do so and as such is not the best choice. Our approach can therefore be considered as a guideline, in order to introduce improved techniques in the future.

B. Maximizing the new modularity

The discussion so far completes our first task of introducing modularity functions that are consistent with

the properties of correlation matrices. Our second task is that of incorporating the above definition(s) into community detection algorithms that seek to maximize the modularity. Below, we start by briefly mentioning the algorithms we adapted in order to search for the optimal partition (more extended descriptions are in the Appendix) and then prove an important property of the optimal partition itself, namely, the fact that its communities are internally positively correlated and mutually negatively correlated.

1. Redefining three community detection algorithms

Given our new definition of modularity in Eq. (36), we cannot directly apply the traditional optimization algorithms devised for graphs, since the majority of these algorithms rely, in some way or another, on the properties of the original network-based definition of modularity, where the degrees of nodes are used to construct the null model. For this reason, we selected three of the most popular network-based community detection algorithms and reformulated them to be compatible with time-series data and our new definition of modularity. The three algorithms we selected are known as the Potts (or spin-glass) method [42,50], the Louvain method [51], and the spectral method [52]. Note that even if these techniques are customarily referred to as “methods,” they can actually be considered as three different algorithms implementing the same method of modularity maximization. Since the appropriate redefinition of these algorithms can require quite technical discussions, it is described in the Appendix.

We note that there exist many modularity maximization algorithms, some of which may already be much better suited to our definition of modularity. However, we wanted to choose popular algorithms whose original specifications required varying levels of rework, ranging from verification of its suitability to accommodate time-series-based modularity to modifications of the underlying tenets of the algorithm itself.

These choices make it possible to illustrate further differences between network-based and correlation-based community detection problems. The reader is again referred to the Appendix for a detailed discussion of these differences.

2. Identifying anticorrelated communities

We now prove the result that the partition maximizing the modularity (whichever method is used to search for it) is characterized by positive intracommunity (filtered) correlations and negative intercommunity (filtered) correlations.

Let us first define the “renormalized” intercommunity correlations (also see the Appendix),

$$\tilde{C}_{AB}^{(l)} \equiv \sum_{i \in A} \sum_{j \in B} C_{ij}^{(l)}, \quad (39)$$

where the notation $i \in A$ indicates that the node i belongs to the community A , and the sum is over all such nodes. Now, assume that we have identified the optimal partition maximizing the modularity, and consider the modularity change ΔQ_l that would be obtained by further merging two different communities of the optimal partition, say, A and B . From Eq. (36), we can write this change as

$$\begin{aligned} \Delta Q_l &= [\tilde{C}_{AA}^{(l)} + \tilde{C}_{BB}^{(l)} + \tilde{C}_{AB}^{(l)} + \tilde{C}_{BA}^{(l)}] - [\tilde{C}_{AA}^{(l)} + \tilde{C}_{BB}^{(l)}] \\ &= 2\tilde{C}_{AB}^{(l)}. \end{aligned} \quad (40)$$

The above change cannot be positive, otherwise merging A and B would further increase the modularity, which is impossible since A and B are communities of the optimal partition. Therefore, $\Delta Q_l \leq 0$, which also implies $\tilde{C}_{AB}^{(l)} \leq 0$. On the other hand, for every community A of the optimal partition, we must have $\tilde{C}_{AA}^{(l)} \geq 0$, otherwise A would give a negative contribution to the modularity, which is impossible since the partition where all nodes of A are isolated communities would have higher modularity than the optimal partition. Taken together, these considerations imply that

$$\tilde{C}_{AB}^{(l)} \begin{cases} \geq 0 & \text{if } A = B \\ \leq 0 & \text{if } A \neq B. \end{cases} \quad (41)$$

The above result follows simply from the maximization of Eq. (36) and will be confirmed empirically in Sec. V D.

Our algorithms effectively partition the network into mutually anticorrelated communities of positively correlated time series, where it is intended that the term “(anti) correlated” refers to the residual correlations remaining after applying the filtering procedure defined by Eq. (37). For this reason, we will sometimes use the term “residually (anti)correlated” when referring to the sign of filtered correlations. As we will discuss in more detail in Sec. V D, this property has important consequences for portfolio optimization and risk management.

C. Multiresolution community detection

We now discuss the problem of introducing an appropriate multiresolution method. As we mentioned, one way to resolve a hierarchical community structure in ordinary networks using a modularity-based community detection algorithm is by introducing a resolution parameter ϕ as in Eq. (20). We have already noted, in our discussion of Eq. (28), that the same operation would not cluster correlation matrices appropriately if applied to the naive null model appearing in Eq. (25). The same kind of limitation persists if we introduce a resolution parameter multiplying any of the three improved null models $\langle \mathbf{C} \rangle_l$ defined in Eqs. (36) and (37). While the range of any observed correlation coefficient C_{ij} is $[-1, +1]$ by construction, a

resolution parameter would unreasonably map the range of the expected correlation $\langle C_{ij} \rangle$ to $[-\phi, +\phi]$. Similarly, since the null correlation matrices $\langle \mathbf{C} \rangle_l$ we introduced are obtained from the eigencomponents of the observed correlation matrix \mathbf{C} , rescaling them by ϕ is equivalent to an overall rescaling of the corresponding eigenvalues of \mathbf{C} , which is again an unjustified operation.

Given the above limitations, which indicate a lack of theoretical foundation for resolution parameters in the case of correlation matrices, we introduce a completely different multiresolution approach that is specifically designed for multiple time series and has no counterpart in network analysis. After running one of our newly introduced community detection algorithms on the original empirical correlation matrix \mathbf{C} , for each community of size s in the optimal partition, we consider the corresponding $s \times s$ submatrix \mathbf{C}_* of \mathbf{C} . For this submatrix, we define the three null models $\langle \mathbf{C}_* \rangle_l$ as discussed in Sec. IV A for the original matrix \mathbf{C} . By running our community detection algorithms recursively inside each of the communities, we can thus resolve subcommunities within communities. Iterating this procedure identifies a hierarchical community structure, if present. Within each community, the procedure stops automatically when it resolves no further subcommunities.

At each iteration, the “noise” component $\mathbf{C}_*^{(r)}$ will have the same interpretation as when it is identified on the entire correlation matrix since \mathbf{C}_* is the submatrix of the original matrix \mathbf{C} and *not* of the filtered matrix $\mathbf{C}^{(l)}$ defined in Eq. (37), so it still contains the node-specific noise component (the reason why we do not consider the submatrix of $\mathbf{C}^{(l)}$ is because, as we mentioned, the latter may not be a proper correlation matrix [31–34] and thus cannot be filtered further using RMT). The global mode $\mathbf{C}_*^{(m)}$ is now interpreted as the community mode, i.e., a common factor influencing all the time series within that particular community. This will now include both the system-wide mode $\mathbf{C}^{(m)}$, restricted to the subspace relative to \mathbf{C}_* , that would be identified on the entire matrix \mathbf{C} (e.g., in the case of financial time series, the market mode) *and* a genuinely community-specific mode not shared with the time series in other communities. Different communities are therefore possibly characterized by different community modes, and the fact that both this mode and the restriction of the global mode are filtered out is precisely what allows the algorithm to resolve deeper hierarchical modules. Finally, the group component $\mathbf{C}_*^{(g)}$ represents the effect of subgroups nested within the specific community, if present.

It should be noted that the original correlation matrix \mathbf{C} typically has large dimensionality (large N), a property ensuring that the results of RMT, in particular, the expected eigenvalue distribution appearing in Eq. (11), hold to a satisfactory level. However, when considering smaller subcommunities, RMT becomes less reliable because

Eq. (11) no longer holds for small sets of nodes. For this reason, for small submatrices (low-dimensional \mathbf{C}_*), it is preferable to determine the eigenvalues λ_{\pm} not via Eq. (12) but by randomly shuffling the temporal increments of the original time series and constructing the corresponding spectrum as shown in Fig. 1.

We conclude by noting that, for the particular case of multiple time series, there is another multiresolution character, which can be attached to the problem of community detection, namely, the fact that different communities can, in principle, be obtained for different choices of the initial temporal resolution, i.e., for different choices of the frequency of the original time series (e.g., second, minute, or daily returns). Note that this notion of temporal resolution is specific to correlation matrices and has no analogue in the ordinary problem of community detection in networks. It is also not necessarily attached to an idea of hierarchy, in the sense that we do not expect, e.g., communities obtained at higher frequencies to be necessarily nested within communities obtained at lower frequencies (even if this can reasonably happen in some cases). To distinguish this specific notion from the usual one of multiresolution community detection, we will refer to it as the multifrequency problem and address it separately in Sec. VI.

D. Benchmarking our methods

Before applying our methods to the analysis of real correlation matrices, we ran a series of tests confirming that we can correctly detect correlated sets of time series in controlled benchmark cases. Our benchmarks consist of heterogeneously sized communities of time series that are internally correlated and additionally display varying levels of noise and global signal (market mode). The reason why we consider heterogeneous community sizes is because this is the more challenging case where we showed the naive method to display a higher bias (see Sec. III C).

We constructed these benchmarks by first choosing the number N of time series, the number c of communities, and the desired number n_A of time series in each community A , such that $\sum_{A=1}^c n_A = N$ (as in Sec. III C). Then, we generated c random and uncorrelated time series (with $T > N$) with values $\gamma_A(t)$ (where $1 \leq A \leq c$) drawn independently from a normal distribution with zero mean and unit variance. Next, we created n_A identical copies of the A th time series, for all A . To each of the resulting N time series, each labeled by an index i , we added a local noise $\beta_i(t)$ (a new normally distributed random variable with zero mean and unit variance, independent of all the other ones) multiplied by a “noise parameter” $\nu \geq 0$ and a global signal $\alpha(t)$ (again, an independent normally distributed random variable with zero mean and unit variance) multiplied by a market-mode parameter $\mu \geq 0$. This resulted in a set $\{Y_1, \dots, Y_N\}$ of N time series with values

$$y_i(t) = \mu \cdot \alpha(t) + \nu \cdot \beta_i(t) + \gamma_A(t) \quad i \in A, \quad (42)$$

$$A = 1, c.$$

Note that this procedure is similar to the so-called ‘‘factor models’’ used in financial analysis [5,29,30,53–55]. The time series $\{Y_1, \dots, Y_N\}$ were further standardized to obtain a final set $\{X_1, \dots, X_N\}$ of N time series, each with zero mean and unit variance, in compliance with the general prescription mentioned in Sec. II.

We generated several benchmarks according to the recipe described above, for various choices of N , c , $\{n_A\}$, μ and ν . In general, when $\mu = \nu = 0$ the benchmark is similar to the ideal one described in Sec. III C: The communities are completely correlated internally (all the time series in the same communities are identical) and uncorrelated with the time series in other communities. This results in a benchmark partition $\vec{\sigma}^*$ such that, for infinite time series, $C_{ij} = \delta(\sigma_i^*, \sigma_j^*)$. However, for finite (but still such that $T > N$ as prescribed by random matrix theory; see Sec. II D) time series, C_{ij} will be affected by noise. As μ and ν increase, additional noise will be generated and the

community structure will be more difficult to detect. If $\mu = 1$ ($\nu = 1$), then the amplitude of the global mode (local noise) is the same as that of the community signal. Therefore, when μ and/or ν approach or exceed 1, the community detection problem becomes more challenging. Still, the ambition of our method is that of correctly identifying the benchmark partition $\vec{\sigma}^*$ even in this hard regime.

In Fig. 4, we show nine benchmarks, organized in a 3×3 table with different combinations of values for μ and ν . In all these cases, the communities to detect are the same set of $c = 8$ heterogeneously sized communities shown previously in Fig. 2(c). The color maps show the values of the entries of the filtered correlation matrix $\mathbf{C}^{(g)}$ defined in Eq. (17), i.e., the residual correlations obtained after removing the noise and market-mode components. It can be seen that, even for values of μ and ν exceeding 1, the filtered matrices always display a clear block-diagonal structure with a visible contrast across diagonal and off-diagonal blocks.

In all of these benchmarks, we confirmed that, using the corresponding modularity $Q_3(\vec{\sigma})$ defined in Eq. (35), our

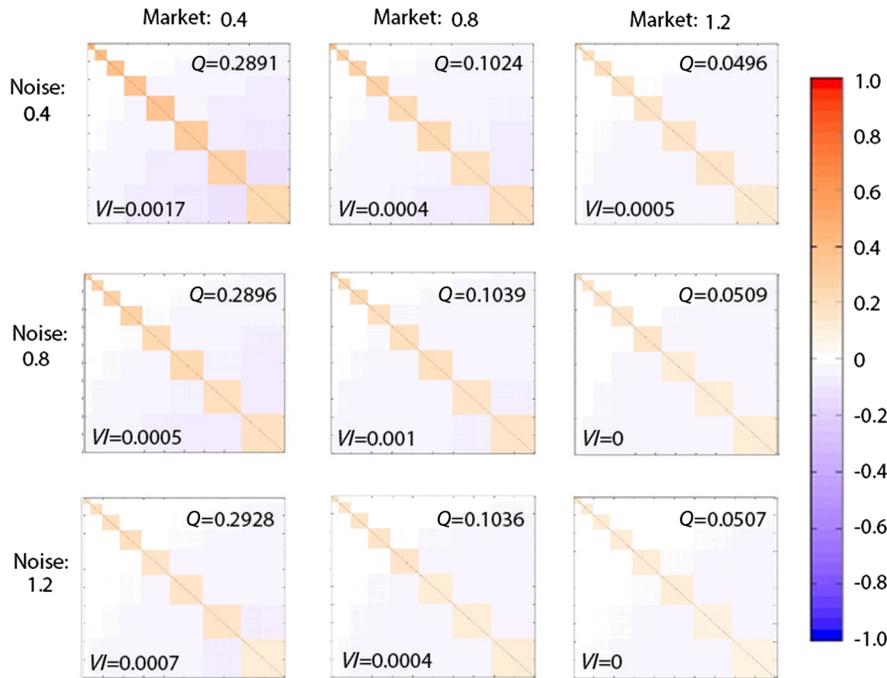


FIG. 4. Performance of our method on nine benchmark sets of correlated time series with varying levels of noise (ν) and market-mode (μ) components. For each combination of μ and ν , $N = 1000$ time series of length $T = 50000$, partitioned into $c = 8$ communities (always containing 35, 60, 85, 110, 140, 165, 190, and 215 time series, respectively), were initially generated according to Eq. (42). Then, the 1000×1000 correlation matrix \mathbf{C} was calculated. The heat maps in this figure show the values of the entries of the filtered matrix $\mathbf{C}^{(g)}$ defined in Eq. (17) and obtained by removing the noise and market-mode components from the original correlation matrix. Blocks along the diagonal represent the residual correlations within each community, while off-diagonal blocks show the residual negative cross-correlations among communities. Our method, using the Potts algorithm (see the Appendix) here to maximize the modularity $Q_3(\vec{\sigma})$ defined in Eq. (35), was always able to correctly identify the target communities, even for values of μ and ν exceeding 1. This is indicated by the value of VI (averaged over 10 runs of the community detection algorithm) calculated between the ‘‘true’’ and the detected partition in each benchmark. The average (over multiple runs) maximum modularity value $Q_3(\vec{\sigma}^*)$ is also shown in each case.

method succeeded in detecting the correct partition $\vec{\sigma}^*$. We quantitatively measured the performance of our method in terms of a metric known as variation of information (VI) [56,57], which measures the entropy difference between two partitions of the same network, providing a rigorous way for us to quantify the similarity between the true partition and the one identified by our method. More precisely, VI involves the use of Shannon’s entropy to measure the amount of uncertainty that exists across the set of communities of two different partitions of the same network. It provides a quantitative measure of the difference between two partitions, a normalized value where zero implies the two partitions are completely identical and 1 implies that they are completely unrelated. As can be seen from Fig. 4, the values of VI (averaged over multiple runs of the community detection algorithm) are zero or extremely small, indicating a perfect or almost perfect performance of the method.

The average (over multiple runs) maximum modularity value $Q_3(\vec{\sigma}^*)$ obtained in the above benchmarks is also illustrated in Fig. 4. Lower values of the modularity imply that the network as a whole is more homogeneous in its construction, to the extent that the detected communities exhibit only a relatively weak increase in their collective correlation, above the ambient level. As expected, we see that the modularity decreases for increasing levels of market mode. Increasing levels of noise, however, do not have such a strong effect since noisy time series tend to diminish the strength of the intracommunity correlations, which enter in both the numerator and denominator of the modularity. In contrast, the market mode has a significant impact on the intercommunity correlations, which primarily end up only in the denominator of the modularity, hence the observed decrease in modularity with an increase in market mode. The corresponding low values of the modularity confirm what we had anticipated about the effects of Eq. (38). We should bear these effects in mind when interpreting the (low) values of the modularity arising from the partition of real financial time series, where the market mode is very strong. The fact that our method correctly identifies the benchmark partitions even for strong market mode (and low resulting modularity) makes us confident that it will also properly detect the community structure of real markets.

V. THE MESOSCOPIC ORGANIZATION OF REAL FINANCIAL MARKETS

Having redefined the modularity consistently with the properties of correlation matrices and appropriately reconfigured three different techniques for optimizing it, we are now in a position to apply our methodology to a variety of real-world data sets and evaluate the quality of the results. In particular, we will apply our three algorithms and the null model expressed in Eq. (35) to time series representing stock prices from a variety of stock indexes that span

multiple industries and multiple countries. We first obtained static results, including the multiresolution community structure as introduced in Sec. IV C, using time series of log-returns of daily closing prices for all three indexes. These results are shown in this section. Then, we considered different temporal (frequency) resolutions and studied the time dynamics of community structure. These additional results are described in Secs. VI and VII, respectively.

A. Data and preprocessing

The indexes we used are the S&P 500 (US Large Cap. Stocks), the FTSE 100 (British Large Cap.), and the Nikkei 225 (Japanese Large Cap.). For each of these indexes, we considered a period of 2500 trading days, corresponding to approximately ten years of market activity, from 2001Q4 to 2011Q3. We selected all stocks for which complete data are available during this period. This resulted in the selection of 445 S&P stocks, 78 FTSE stocks, and 193 Nikkei stocks. All these stocks are classified within the Global Industry Classification Standard (GICS) [58]. The complete taxonomy can be found online [59]; however, we briefly mention that there are ten top-level “sectors” (see Table I) split into 24 subcategories called “industry groups,” which are in turn divided into 68 “industries.”

It is important to note that, although we would expect stocks within certain industry sectors to be correlated with each other, we do not expect to observe this effect within and throughout all industry sectors. Previous research in the area of stock clustering [4–6,20,60,61] (see also our discussion in Sec. II) has shown some relationships between the industry sectors and clusters of stocks identified by the various methods. We therefore expect to find a certain degree of overlap with this research. However, our choice of null models in conjunction with our tailored community detection algorithms is designed to uncover nontrivial correlations, beyond a direct mapping to industry sectors, such as finding stocks from different industry sectors that tend to move together, and even in opposition to other stocks in their own sector. It is therefore useful to use industry sectors not as a target but as a baseline to highlight important and nontrivial deviations identified by the community detection algorithms.

As with the benchmarks described in Sec. IV D, each set of financial time series was used to initially create a correlation matrix \mathbf{C} that was then filtered to produce

TABLE I. The 10 industry sectors in the Global Industry Classification Standard (GICS), with the color representation used to highlight the sectors in the following figures.

Consumer discretionary:	■	Consumer staples:	■
Energy:	■	Financials:	■
Health care:	■	Industrials:	■
Information technology:	■	Materials:	■
Telecommunication services:	■	Utilities:	■

the matrix $\mathbf{C}^{(g)}$ following the procedure described in Secs. II D and IV A. Each such matrix was then operated on individually by the three community detection algorithms described in Sec. IV. We found that all algorithms always generate very similar partitions. This important result, which, for the sake of exposition, is postponed to Sec. V E, implies that we can refrain from showing the results of every algorithm. For brevity, we will instead select representative exemplars, with the understanding that any one of the algorithms would generate very similar results.

B. Standard approaches

Before showing the main results of our own methodology, as a preliminary study we illustrate what would be obtained by using some of the standard approaches available, in particular, the correlation threshold described in Sec. II A and the community detection built on the network-based modularity described in Secs. III B and III C.

1. Asset graph from Fisher-transformed correlations

As we discussed in Sec. II A, imposing a threshold on the entries of a correlation matrix \mathbf{C} allows us to obtain an asset graph where links connect the more strongly correlated pairs of stocks [4,19,20]. In Fig. 5, we show the effect of this procedure on our S&P 500 data. Rather than showing the results for multiple choices of the threshold, we used a rough criterion to select a unique threshold that would, in principle, correspond to a standard level of statistical significance. This criterion is as follows.

As well known in statistics [62], one can easily show that, under the null hypothesis, two time series X_i and X_j of length T representing T realizations of two independent and normally distributed random variables, the quantity

$$z_{ij} \equiv \operatorname{artanh} C_{ij} = \frac{1}{2} \ln \frac{1 + C_{ij}}{1 - C_{ij}} \quad (43)$$

(where C_{ij} is the sample correlation coefficient) is distributed as a normal variable with zero mean (representing the population correlation coefficient in the case of independent variables) and standard error

$$\sigma = (T - 3)^{-1/2}. \quad (44)$$

In other words, under the above null hypothesis, we expect a concentration of values of z_{ij} around zero, with standard error σ .

In order to detect significant deviations from the null hypothesis, one may select a threshold τ such that only the values outside τ standard errors, i.e., $|z_{ij}| > \tau\sigma$, are considered as statistically significant. This means that one can select a threshold $z_\tau \equiv \tau\sigma$ for z_{ij} . In terms of the correlations C_{ij} , the corresponding critical value is

$$C_\tau \equiv \tanh z_\tau = \frac{\exp\left(\frac{2\tau}{\sqrt{T-3}}\right) - 1}{\exp\left(\frac{2\tau}{\sqrt{T-3}}\right) + 1}. \quad (45)$$

A suitable choice of the value of τ can be used as a threshold to project the correlation matrix into an asset graph at the corresponding significance level: Specifically, one can draw a link only if $|C_{ij}| > C_\tau$. The advantage of introducing the above criterion is that, at least in principle, it associates a precise statistical significance level to any value of the threshold (there are, however, various problems with this approach, as we briefly comment later). This makes it possible to select a unique threshold value corresponding to a standard accepted level of significance.

We used the above approach as a rough criterion to select an indicative threshold, choosing $\tau = 2$ so that only the correlations lying two standard deviations away from the null hypothesis are, in principle, retained. The resulting asset graph for the stocks of the S&P 500, plotted in Fig. 5, was visualized using a clustered rendering [63] of all the stocks that do not end up completely isolated after the filtration, according to the Fruchterman-Reingold [64] force-based algorithm. As expected, we immediately see a significant correspondence between groups of densely connected nodes and industry sectors. However, there is no linear relationship between the attractive and repulsive forces defined by the graph-drawing algorithm and the contribution of the corresponding correlations to the modularity. As such, the visualization of the graph cannot be directly used to partition the network into communities.

Moreover, it should be noted that the approach we have used to define a threshold has two main theoretical disadvantages: First, it assumes normally distributed log-returns (while it is well known that real log-return distributions are fat-tailed [65,66]); second, it does not introduce multiple hypothesis test corrections. A more rigorous way to statistically validate links in a correlation-based network would be that of using numerical bootstrapping methods such as the one considered in Ref. [67].

In any case, since no search over the space of possible partitions is performed, the asset-graph method cannot identify communities of stocks that are more strongly correlated internally than with the rest of the market. As we anticipated in Sec. II A, this leaves the problem we started with unsolved. We also recall from Sec. IV B 2 that our methodology detects residually anticorrelated communities. This property, which we will illustrate in Sec. V D for the data considered here, cannot be achieved by any threshold-based method or any of the other available methods we described in Sec. II.

2. Naive application of community detection

As another baseline reference, in Fig. 6 we show the result of applying, to the same S&P 500 data, the

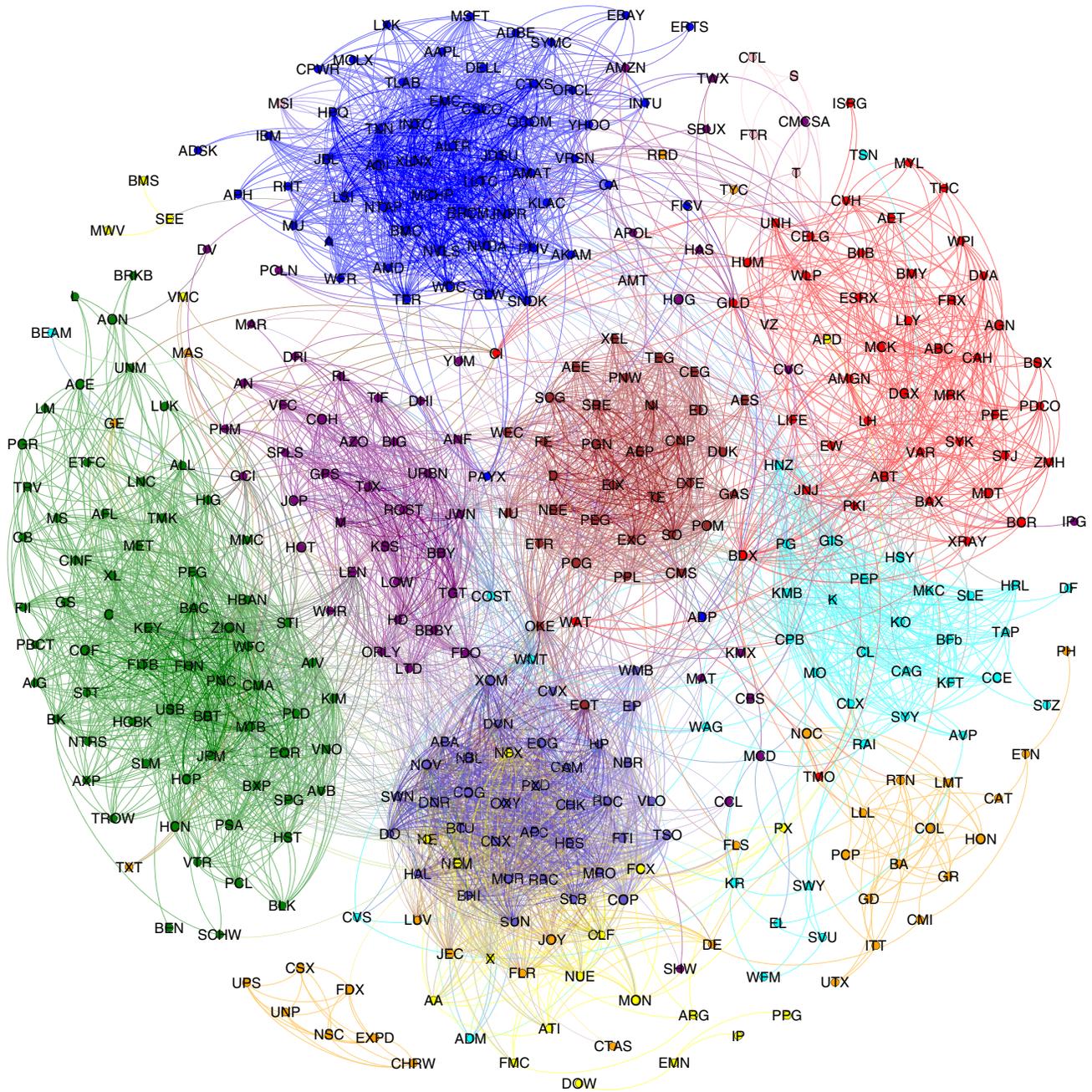


FIG. 5. Asset graph for the S&P 500 (log-returns of daily closing prices from 2001Q4 to 2011Q3). The network is generated from the correlation matrix of the constituent stocks, after taking the Fisher transform and setting a threshold at 2 standard deviations. The color of each node represents the industry sector to which that stock belongs (see Table I). The force-based layout clearly indicates the existence of strong connections between stocks of the same industry sector; however, this approach (like any other threshold-based approach) cannot identify communities of stocks that are internally more correlated than with the rest of the market, and mutually anticorrelated.

community detection described in Sec. III B, i.e., by treating the correlation matrix \mathbf{C} as a weighted network and running an ordinary (network-based) community detection algorithm [13,14]. We see that the resulting trivial community is a single one spanning the entire set of stocks. In such a case, the pie chart depicting the community merely illustrates the distribution of industries within the S&P 500. The same result is obtained if one uses

the correlation-based modularity $Q_1(\vec{\sigma})$ defined in Eq. (33) in terms of the null model $\langle \mathbf{C} \rangle = \mathbf{1}$ (i.e., assuming that all time series are completely independent and of infinite length).

In the first case, this result is due to the inconsistent structure of the modularity and to the resulting bias of the algorithms used to maximize it, as we discussed in Sec. III B. In the second case, it is due to the inadequacy of

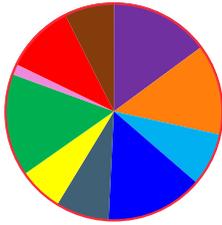


FIG. 6. The trivial, single community containing all stocks of the S&P 500 (log-returns of daily closing prices from 2001Q4 to 2011Q3), obtained by either naively treating the correlation matrix as a weighted network and using the ordinary network-based modularity or, alternatively, using the correlation-based modularity $Q_1(\vec{\sigma})$ (i.e., without filtering the correlation matrix). In both cases, the Louvain algorithm (see the Appendix) has been used. The colors represent different GICS sectors (color legend in Table I) and span an area proportional to the number of stocks in each sector.

the null model defined in Eq. (26) for financial correlations (see Sec. IV A): The community detection algorithm finds only a single community because of the systemic correlation of the market mode affecting all stocks simultaneously.

C. Community detection using our method

We now come to the application of our own methodology described in Sec. IV. In Fig. 7 we show the result of the application, to the same daily S&P 500 data, of the appropriately redefined community detection methods introduced in Sec. IV B, specifically making use of the modularity $Q_3(\vec{\sigma})$ defined in Eq. (35). Since such null models discount both random and market-wide correlations, the community detection algorithms are now able to successfully find correlations that exist in between the microscopic and macroscopic levels. For the S&P 500, the result is a set of five mesoscopic communities whose relative size (the number of nodes in each community) is expressed by the size of the pie chart in the graph. The relative breakdown of the stocks in each community, classified according to their top-level GICS sector (see Table I), is represented by the fraction of the pie chart for that community.

In addition to the communities presented for the S&P 500, in Figs. 8 and 9 we also provide the communities for the FTSE 100 and the Nikkei 225, respectively, again detected using the null model from Eq. (35). As before, the naive community detection would place all stocks into a single community (not shown). For all of these data sets, the values of the maximized modularity $Q_3(\vec{\sigma}^*)$ achieved by the optimal partitions will be shown later in Sec. V E.

While at first glance it may seem as though there is no particular pattern to the community structures in the three markets (as each community contains a plethora of stocks from different industry sectors), a closer look at the industries to which the stocks belong does in fact yield some interesting observations. First and foremost, some of

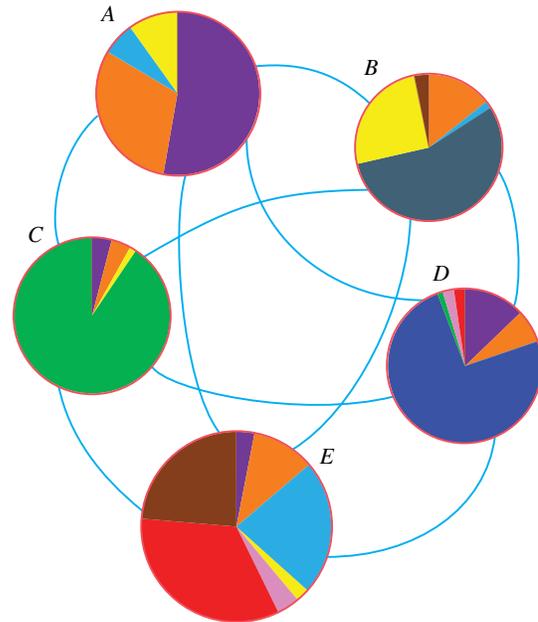


FIG. 7. Communities of the S&P 500 (log-returns of daily closing prices from 2001Q4 to 2011Q3) generated using our correlation-based modularity $Q_3(\vec{\sigma})$ with the Louvain algorithm (see the Appendix). Individual communities are labeled A through E, and the pie chart represents the relative composition of each community based on the industry sectors of the constituent stocks (color legend in Table I). The blue intercommunity link weights are negative, indicating that the communities are all residually anticorrelated. The red circles around each community indicate that the total intracommunity correlations are all positive.

the industry sectors tend to dominate communities, where in some cases 100% of the stocks for a particular industry sector are in the same community, meaning that, on average, over the past ten years, they have all remained correlated. Examples of this include Energy (■, community B), Financials (■, community C), and Information Technology (■, community D) in the S&P; Utilities (■, community A), Health Care (■, community A), Information Technology (■, community C), Telecom. Services (■, community C), and Energy (■, community E) in the FTSE; and finally, Utilities (■, community B), Energy (■, community B), and Consumer Staples (■, community B) in the Nikkei.

There are also instances where top-level sectors are split among different communities according to their subclassification (Industry Group and Industry). This is very interesting because it shows that subgroups of stocks within one sector are often more correlated with a different sector than their own sector.

Other interesting cross-sector correlations can be found too, with Health Care, for example. In the FTSE communities, Health Care stocks (■) are exclusively in community A, whereas in the S&P, they are predominately in community E, with some in community D. Interestingly

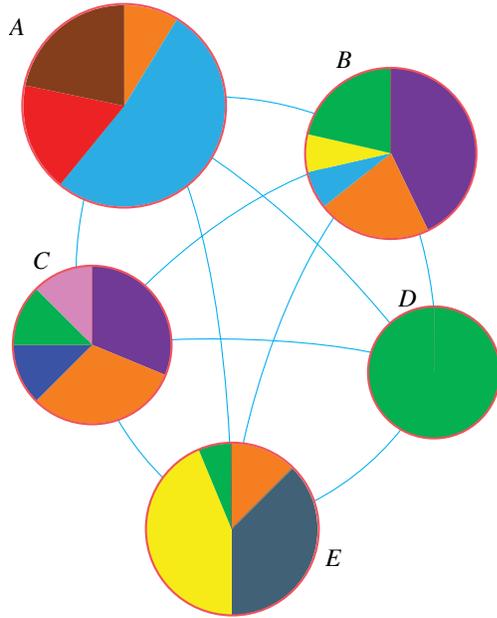


FIG. 8. Communities of the FTSE 100 (log-returns of daily closing prices from 2001Q4 to 2011Q3) generated using our correlation-based modularity $Q_3(\vec{\sigma})$ with the Louvain algorithm (see the Appendix). Individual communities are labeled A through E, and the pie chart represents the relative composition of each community based on the industry sectors of the constituent stocks (color legend in Table I). The blue intercommunity link weights are negative, indicating that the communities are all residually anticorrelated. The red circles around each community indicate that the total intracommunity correlations are all positive.

enough, in the latter case, the *Health Care Technology* industry sector stock from the Health Care sector is in community D, which also happens to be the community containing all of the Information Technology (IT) stocks (■), whereas all of the Pharmaceutical stocks are in community E, which contains the bulk of the Consumer Staples (■) stocks. The reader might, at this point, notice that the FTSE community A containing all Health Care (■) stocks also contains the bulk of the Consumer Staples (■) stocks. It is probably not surprising then to discover that those Health Care stocks are comprised entirely of Pharmaceuticals. We find an identical relationship between Pharmaceuticals and Consumer Staples in community B of the Nikkei 225 as well. Furthermore, the one other Health Care stock, a *Health Care Equipment & Supplies* stock, trades in the same community as the IT stocks. This might not be particularly interesting except for the fact that in the Nikkei, the vast majority of IT sector stocks are subclassified as Electronic Equipment.

One might continue finding interesting trends such as these; however, our purpose is not to glean specific qualitative information regarding financial markets but rather to illustrate how the underlying quantitative information can be ascertained from the raw data, through the appropriate choice of null models in conjunction with the

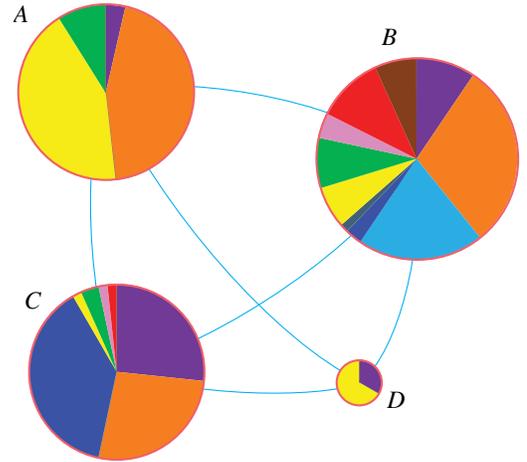


FIG. 9. Communities of the Nikkei 225 (log-returns of daily closing prices from 2001Q4 to 2011Q3) generated using our correlation-based modularity $Q_3(\vec{\sigma})$ with the Louvain algorithm (see the Appendix). Individual communities are labeled A through D, and the pie chart represents the relative composition of each community based on the industry sectors of the constituent stocks (color legend in Table I). The blue intercommunity link weights are negative, indicating that the communities are all residually anticorrelated. The red circles around each community indicate that the total intracommunity correlations are all positive.

process of community detection. The most important result of this process is the successful identification of mesoscopic communities of correlated stocks that are irreducible to a standard sectorial taxonomy and also anticorrelated with each other, as we now discuss.

D. Residually anticorrelated communities and portfolio optimization

The age-old proverb, “*Don’t put all your eggs in one basket,*” could never be more insightful than when deciding how to invest one’s money. Entire departments of almost every investment bank, insurance firm, and hedge fund are dedicated to picking the right baskets for their customers’ nest eggs. This process is often referred to as portfolio optimization (or asset allocation) and involves optimizing the way in which a sum of money is divided up between a variety of financial instruments such that one maximizes the return for a given risk or, alternatively, minimizes the risk for a given return. According to modern portfolio theory (MPT) [68–70], which is widely used in the financial world to calculate asset allocations, one of the most effective ways to accomplish this is through diversification, that is, to select groups of assets which are as uncorrelated as possible, or even anticorrelated.

Clearly, we can identify numerous parallels between MPT and our community detection method. As we anticipated in our proof of Eq. (41), a key property of the correlation-based modularity is that its maximization will identify mutually anticorrelated groups of time series

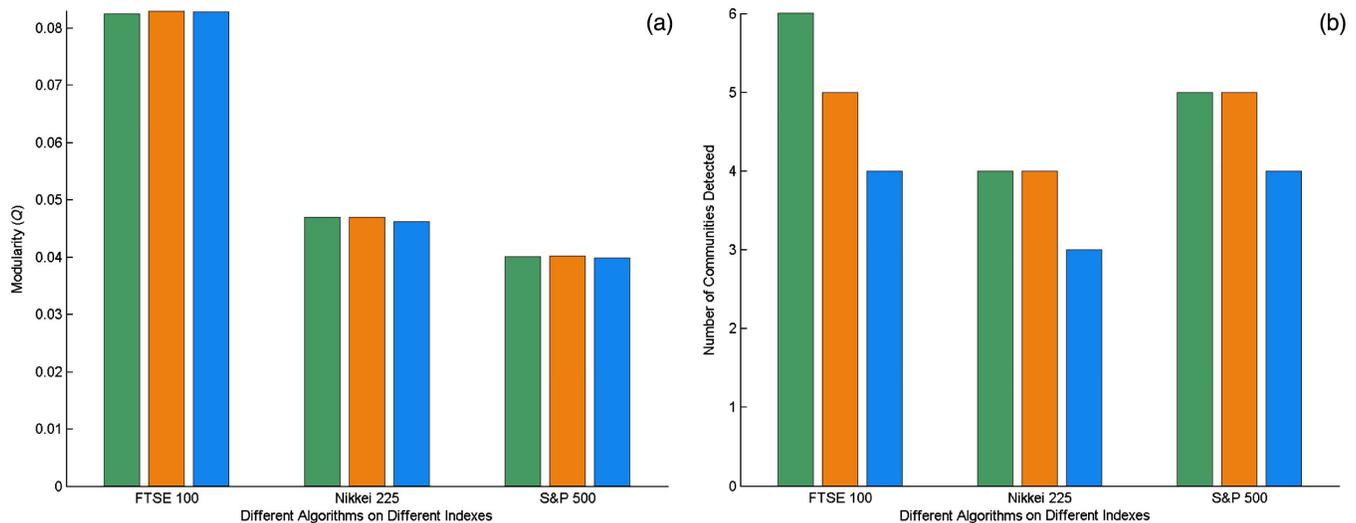


FIG. 10. Maximized modularity values $Q_3(\vec{\sigma}^*)$ (a) and number of detected communities (b) for each of the three markets (log-returns of daily closing prices from 2001Q4 to 2011Q3). The green bar is the Potts algorithm, the orange bar is the Louvain algorithm, and the blue bar is the spectral algorithm.

(where anticorrelations are intended as residual, if some filtering has been applied). Indeed, in Figs. 7, 8 and 9, all the links connecting different communities have negative weights, i.e., all communities are mutually anticorrelated.

The (residual) anticorrelations among communities allow us to identify combinations of stocks, which on top of the overall market mode and purely random fluctuations, move in opposition to each other. Recalling from Eqs. (A6), (A8) and (A9) that

$$\tilde{C}_{AB}^{(l)} = \text{Cov}[\tilde{X}_A, \tilde{X}_B] - \langle \text{Cov}[\tilde{X}_A, \tilde{X}_B] \rangle_l \quad (46)$$

where $\tilde{X}_A \equiv \sum_{i \in A} X_i$, we obtain a practical recipe to construct a set $\{\tilde{X}_A\}$ of community-specific indexes (each built as the sum of the time series of the stocks within a community) such that, as follows from Eq. (41),

$$\text{Cov}[\tilde{X}_A, \tilde{X}_B] < \langle \text{Cov}[\tilde{X}_A, \tilde{X}_B] \rangle_l \quad \text{if } A \neq B. \quad (47)$$

In other words, the two indexes are residually less correlated with each other than expected under the null model; i.e., their mutual filtered correlations are negative. This is a desirable trait from the point of view of risk management and portfolio optimization.

E. Comparative analysis of the three algorithms

In Fig. 10 and in Table II, we show results supporting a claim that we anticipated at the beginning of this section, i.e., the fact that the three algorithms we introduced in Sec. IV B identify a very similar community structure on the data we considered. This finding makes the results shown so far quite robust under changes of the protocol used to derive them.

In particular, in Fig. 10 we show the value of the maximized modularity $Q_3(\vec{\sigma}^*)$ and the number of detected

communities as a result of running all of our three algorithms on the filtered correlation matrices for the S&P 500, the Nikkei 225, and the FTSE 100. We recall from the discussion following Eq. (38) and from the benchmarks studied in Sec. IV D that, unlike the corresponding problem in network analysis, our choice of C_{norm} implies very small values of the maximized modularity, even in the presence of well-defined communities, when the market mode is strong. So the small values of $Q_3(\vec{\sigma}^*)$ shown in Fig. 10(a) do not imply a poor or weak community structure. It can be seen that all three algorithms perform very closely in terms of the maximized modularity value they achieve. Similarly, if we compare the number of communities found by the three

TABLE II. Comparison of the relative variation of information between the optimal partitions found by all algorithms, for the S&P 500, the Nikkei 225, and the FTSE 100. The data are log-returns of daily closing prices from 2001Q4 to 2011Q3.

S&P 500	Potts	Louvain	Spectral
Potts	0	0.019	0.09
Louvain	0.019	0	0.08
Spectral	0.09	0.08	0

Nikkei 225	Potts	Louvain	Spectral
Potts	0	0.007	0.04
Louvain	0.007	0	0.04
Spectral	0.04	0.04	0

FTSE 100	Potts	Louvain	Spectral
Potts	0	0.11	0.11
Louvain	0.11	0	0.05
Spectral	0.11	0.05	0

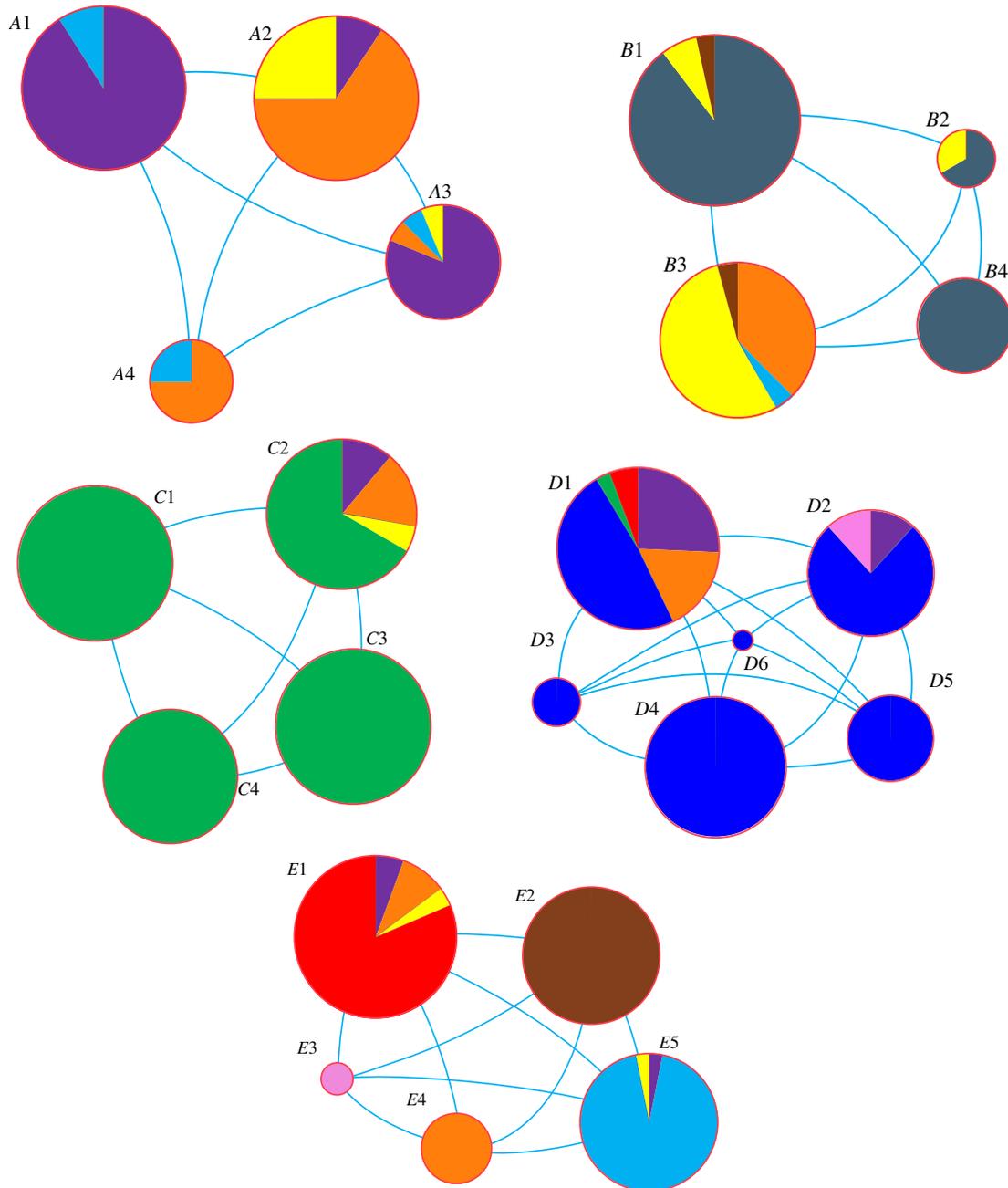


FIG. 11. Our multiresolution community detection method resolves the subcommunity structure of the five communities of the S&P 500 (see Fig. 7). Community *A* mainly comprises Consumer Discretionary and Industrial stocks, *B* all the Energy stocks, *C* all the Finance stocks, and *D* all the IT stocks, while *E* is highly heterogeneous but very well resolved into five separate subcommunities mainly comprising Utilities, Industrial, Health Care, Telecommunication Services, and Consumer Staples stocks, respectively. Besides this relatively predictable partition, we note that Industrials stocks, and to a lesser extent also Materials and Consumer Discretionary stocks, are quite dispersed across different communities. (From cross-correlations of log-returns of daily closing prices from 2001Q4 to 2011Q3; the Louvain algorithm has been used.)

methods (see Fig. 10(b)), we find that the number of communities is quite stable as well.

In Table II, we quantify more rigorously the differences in the composition of the communities detected by the three algorithms, by showing the *VI* (see Sec. IV D) among all pairs of algorithms, for all three indexes. The values are

quite low, indicating that the partitions found by different algorithms are very similar.

F. Hierarchical community structure of the market

We now come to the application of the multiresolution community detection approach we introduced in Sec. IV C.

In the case of financial markets, the community-specific correlation responsible for the modular structure shown so far can be regarded, from the perspective of all stocks within one community, as a “micro market mode.” Just as the market mode discussed previously is responsible for the collective tide of an entire market, a similar force can be extrapolated at the community level. As discussed in Sec. IV C, accounting for this “community mode” in the leading eigenvalue and corresponding eigenvector of the correlation submatrix restricted to an individual community allows us to incorporate its effects, together with those of the overall market mode, into the null model and, again, detect any residual underlying structure.

Figure 11 shows the result of a single layer of recursion into the five communities of the S&P 500 (depicted previously in Fig. 7). Again, we note that the subcommunities are all residually anticorrelated with each other (within each parent community) but maintain an internal positive correlation. Although not obvious from the graph, the subcommunities tend to fall along GICS industry sector lines, with some interesting exceptions, as before.

To call out a few examples, in community *D* (Fig. 11), which contains all of the IT stocks, we see the subcommunities separating along Industry Group and Industry lines [71]. Subcommunity *D5* is comprised of only *Software* stocks, *D4* contains all of the *Semiconductor & Semiconductor Equipment* stocks, and *D2* contains all of the *Internet Software & Services* stocks. Interestingly enough though, *D2* also contains Amazon Inc. and Priceline Inc. from the Consumer Discretionary Sector, which one could argue are quite aligned with the Internet. Continuing the analysis further, we see that in community *C*, the Finance community subcommunity *C2* contains all of the *Commercial Bank* stocks, while *C3* contains all but one of the *Insurance*

companies. *C4* is exclusively *Real Estate Investment Trusts (REITs)* and accounts for all of them. Similar partitions can be seen in the other subcommunities, and further recursion into these communities produces still further separation, close to but not exactly in line with the GICS classification.

Figure 12 depicts the hierarchical nature of the S&P 500 to three layers deep. The process can be continued until no single community can be partitioned further into any combination of two or more sets that are anticorrelated with each other. For instance, community *E* (Fig. 11), which contains a variety of stocks from various GICS sectors, separates out such that the bulk of the stocks in the different sectors find themselves in their own subcommunity. If we further probe into community *E1*, which contains all of the Health Care stocks, we see that the subcommunities (not shown) fall very closely along Industry lines, with five communities each comprised predominantly of *Pharmaceutical*, *Biotechnology* and *Life Science Tools*, *Health Care Providers & Services*, *Health Care Equipment & Supplies*, and everything else, respectively. Albeit interesting, these results invite inspection of the stocks that end up in the “everything else” community. These stocks were deemed to be correlated with the other Health Care stocks, when they were all placed in community *E1*, and they include McGraw Hill Inc., H&R Block, and Waste Management Inc. None of these stocks immediately stands out as being fundamentally related to Health Care.

Similar outliers exist in the other communities as well. It may well be the case that there is good reason for their association, for example, a shared parent company, sizable investment, common board members, or some other significant relationship, or it may be purely coincidental. Gaining a better understanding of this takes us to our next line of experimentation.

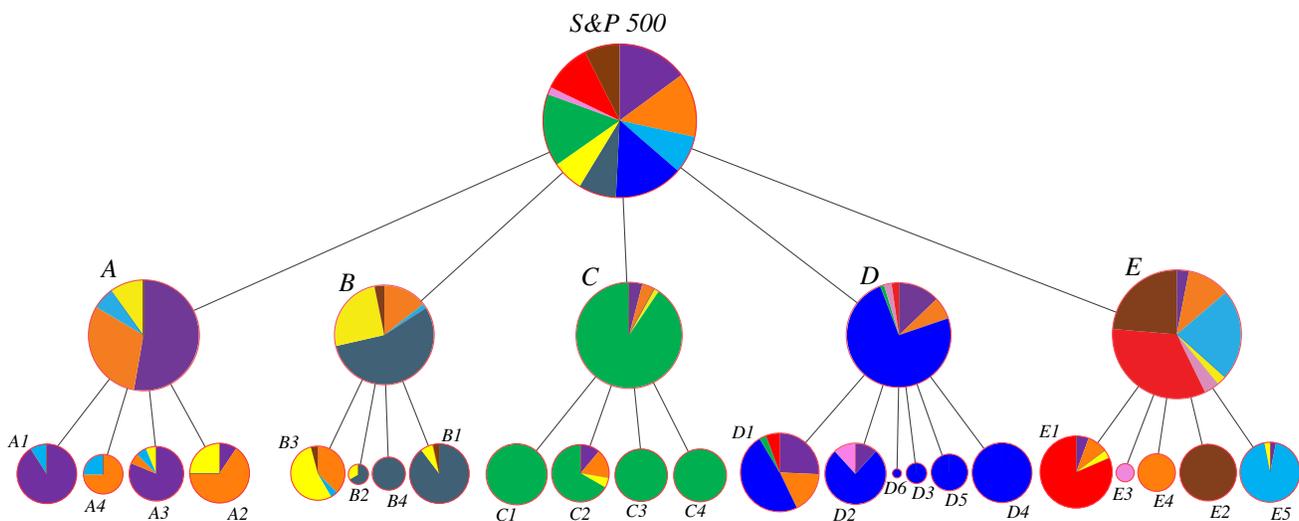


FIG. 12. Multiresolution community detection reveals the hierarchical structure of the communities of the S&P 500 (from cross-correlations of log-returns of daily closing prices from 2001Q4 to 2011Q3). The dendrogram gives an alternative, combined representation of Figs. 6, 7 and 11. (The modified Louvain algorithm has been used.)

VI. MULTIFREQUENCY COMMUNITY DETECTION

Having examined the mesoscopic structure of a set of financial markets, one might be curious as to whether that structure is specific to the chosen frequency of the original time series. In other words, one can check whether the same communities would be retrieved if the returns that comprised the original time series were calculated every minute, every half hour, or every two days. To answer this “multifrequency” community detection problem, in this section, we evaluate the robustness of partitions at a variety of temporal resolutions.

A. Multiple-frequency data

In order to maintain consistency with the results previously described in this paper, we use the same time frame but, instead of working with daily log-returns, we created new data using minute log-returns for the S&P 500 stocks. This process has the initial effect of greatly increasing the amount of data being used: From 2500 data points per stock for the daily returns to approximately 900,000 for the minute returns. In order to accommodate some missing data from the minute returns, we had to reduce the set of 445 to 413 stocks, noting that the removed stocks were relatively evenly distributed across the top-level sectors of the GICS, so as not to deplete any one particular sector. With the minute return time data of these 413 stocks, we created nine new sets of time series, corresponding to a variety of different resolutions Δ_t spanning the same ten-year period:

$$\Delta_t \in \{1, 5, 10, 15 \text{ \& } 30 \text{ mins, } 1 \text{ hour, } 0.5, 1 \text{ \& } 2 \text{ days}\}.$$

For example, the 5-min data were created by taking the price of every stock every 5th minute throughout the day. From these nine sets of time series, we then proceeded in the same fashion as was previously described for daily return data, creating correlation matrices and leveraging RMT filtering to produce the respective null models.

B. Robustness over multiple frequencies

To measure the effects of resolution, we applied all three of the community detection algorithms discussed above to all nine data sets, yielding various values for the modularity $Q_3(\vec{\sigma}^*)$ of the partition (see Fig. 13). Since there can be multiple peaks within a modularity landscape [72], all yielding the same value of $Q_3(\vec{\sigma}^*)$ but exhibiting different community structures, we use VI (see Sec. IV D) as a measure of the difference between the partitions. Since VI is a comparative measure, we (arbitrarily) use the community structure previously ascertained from the daily returns as the point of reference. Thus, as can be seen in Fig. 14, the VI for the 1-day returns is 0 by construction, indicating perfect similarity, whereas the community structure for every other resolution shows some level of deviation.

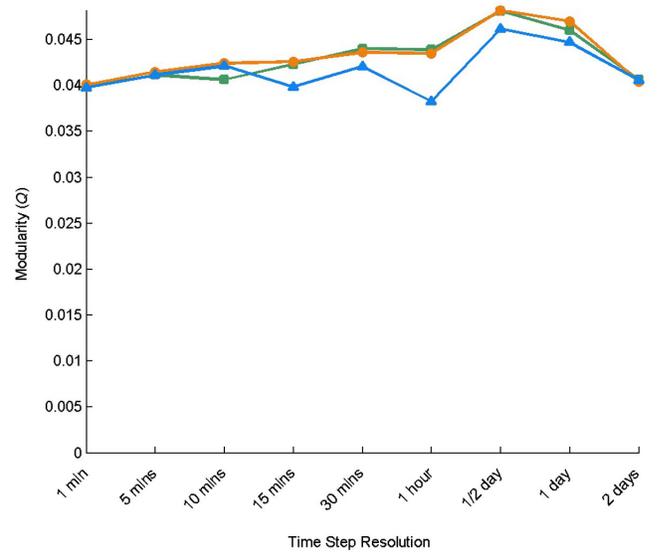


FIG. 13. Multifrequency analysis of the modularity $Q_3(\vec{\sigma}^*)$ for the different methods, as the resolution goes from 1-min intervals to 2-day intervals. A relatively consistent value of the modularity can be seen across all time-step resolutions. (The Potts algorithm is shown by green squares, the Louvain algorithm by orange circles, and the spectral algorithm by blue triangles.)

Overall, it can be seen from the combination of Figs. 13 and 14 that there exists a considerable amount of consistency between the communities detected at differing resolutions, with the $Q_3(\vec{\sigma}^*)$ values remaining almost

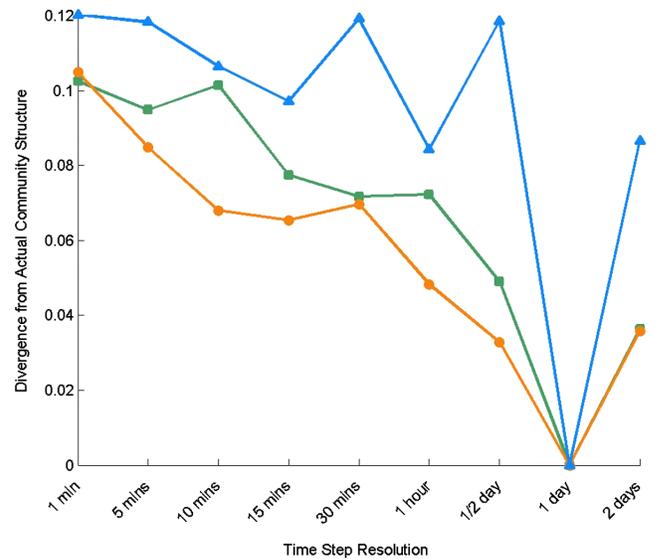


FIG. 14. Multifrequency analysis of the variation of information between each of the nine data sets of different time resolutions and the partition for the data set of daily time steps. It can be seen that all data sets yield partitions quite similar to each other, but there is still slight degradation as the resolution gets finer. (The Potts algorithm is shown by green squares, the Louvain algorithm by orange circles, and the spectral algorithm by blue triangles.)

constant and VI deviating slightly with each resolution interval but indicating, in most cases, no more than a 10% difference between the communities of a particular resolution and those of the 1-day resolution. This means that the correlations between large groups of stocks are not strongly dependent on the resolution of the chosen time step. One might expect to see fluctuations in the variance of stocks at smaller time resolutions, where the more volatile periods of trading (such as market open and market close) are captured. However, since we are dealing with correlation matrices, this variance is normalized away. Moreover, we recall that our definition of C_{norm} in Eq. (38) controls for the varying volatility (variance of the total log-return over all stocks), allowing us to focus solely on the relationships between the stocks themselves.

C. Detection of hard and soft stocks: Overlapping community structure

Although the values of $Q_3(\vec{\sigma}^*)$ and VI do provide reasonable insight into the robustness of community structure at the different resolutions, we take the analysis one step further and examine the communities from the perspective of the individual stocks. In other words, we can further examine the community affiliation of individual stocks at the various resolutions to ascertain the frequency of times any two stocks find themselves in the same community as each other. We show the results of this analysis in Fig. 15, which is a heat map of the different stocks, such that the color of every pair indicates the frequency of co-occurrence in the same community, across all resolutions. For example, if two stocks are always in the same community (unit frequency), then their entry in the heat map is white, while if they are never in the same community regardless of the time step chosen (zero frequency), then their entry is black. Stocks that share a community for some time steps are shades of red (lower frequency) or yellow (higher frequency).

As we can see, the results are in line with the graph of VI in Fig. 14. In other words, the communities tend to consist of a large core of hard stocks that are unwavering over the different resolutions, plus a small amount of soft stocks that fluctuate between communities, presumably giving rise to the 10% fluctuation in community structure observed with the VI analysis. A significant finding is the existence of a group of soft stocks that alternate across the Utilities, Health Care, and Consumer Staples communities, and of another group of soft stocks alternating across the Consumer Discretionary and Financials communities.

It should be noted that our identification of the hard stocks that are, most of the time, part of the core of a community and the soft ones that are instead alternating across communities is a way to take the potentially overlapping nature of communities into account, even if using a nonoverlapping method like modularity maximization. This possibility has no counterpart in the standard

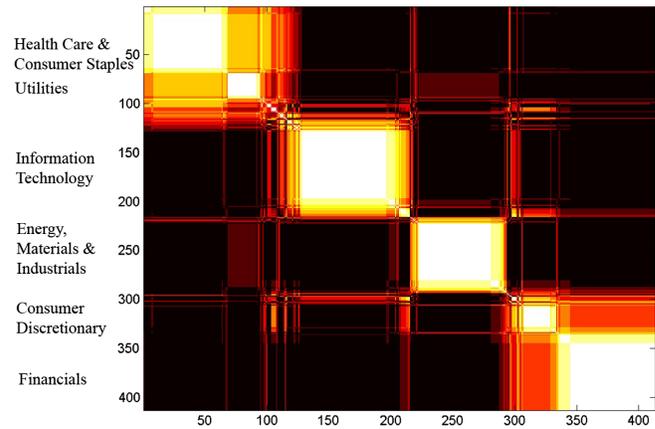


FIG. 15. Multifrequency heat map showing the normalized co-occurrence of different pairs of stocks within the same community, for the same time period but over various temporal resolutions of the original time series (white: unit frequency; black: zero frequency). The stocks have been ordered using simulated annealing to position stocks with a high degree of cross-correlations next to each other. To further inform the graph, the GICS sectors have been specified, emphasizing which groupings of stocks tend to associate with a particular sector. Overall, the blocks of large black and white areas indicate a high degree of coherence of the communities at different resolutions. However, there are two groups of soft stocks, one alternating across Utilities, Health Care, and Consumer Staples, and one alternating across Consumer Discretionary and Financials (produced using the Louvain algorithm).

network-based community detection problem, and it is offered by the intrinsic dependence of correlation matrices on the frequency of the original time series. In what follows, we will use the dynamical evolution of correlations to explore another dimension of variability leading to an alternative way to resolve overlapping communities of multiple time series.

VII. TIME DYNAMICS

When optimizing a portfolio, there is a constant need to choose an adequate period of history from which to try to predict future behavior of the assets in the portfolio. Choosing too short a period will inaccurately bias one's results because extreme events are weighted too heavily. Similarly, choosing too long a history can imply stability where none exists. In general, analyzing the stability of communities over time provides us with reassurance that our models are in fact producing statistically significant results as well as providing insightful information about the data themselves. For example, it is well known in finance that markets become much more globally correlated during periods of economic decline. Stated in the terminology we have been using throughout this paper, they fall more under the influence of the market mode and relinquish the structure provided by the group mode. That being the case, we would expect to see communities lose coherence

during periods in the data set that we know to have been economically troublesome, for example, the tech bubble bursting from 2000–2001 or the subprime lending crisis, 2007–2008.

Since we have shown that the 15-min data set yields very similar communities to the daily data set for the S&P, we can feel reasonably assured that we can use 15-min data instead of the daily data, which will allow us to examine the S&P data set using a sliding time window of two years, corresponding to a ratio $T/N = 6$, and even look at more fine-grained windows, e.g., six months.

A. Two-year window

We now seek to examine the community structure over sequential periods of two years to unearth any anomalies that might exist. We again apply the different methods of community detection using the two-year window time-series sets of the S&P 500 and subsequent null models created using RMT filtering. As we did for our analysis of resolution, here we evaluate the modularity function $Q_3(\vec{\sigma}^*)$ for each period (see Fig. 16) along with the VI (see Fig. 17), where for VI we are comparing each window with the initial two-year window.

As before, we see that all three algorithms perform in a reasonably similar manner. However, unlike our analysis of robustness over different resolutions (which showed little change in community structure or in the modularity), here we see that $Q_3(\vec{\sigma}^*)$ fluctuates over the different windows. We recall again that, as we mentioned in our discussion

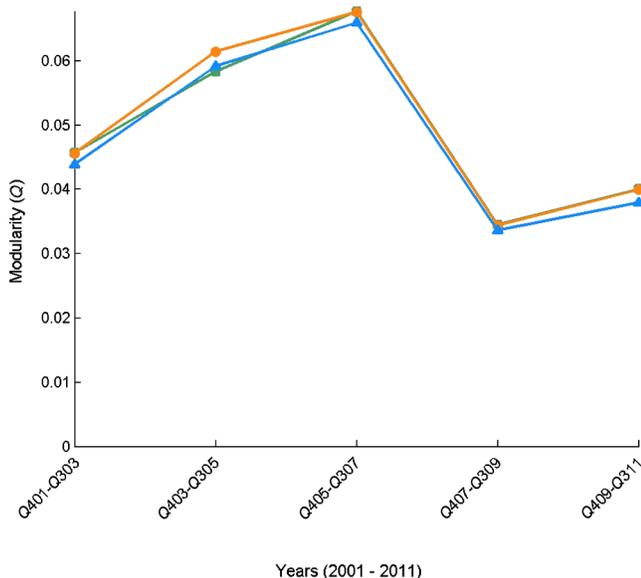


FIG. 16. Temporal trend in the values of $Q_3(\vec{\sigma}^*)$ for the different methods over a two-year sliding window spanning the time frame from October 2001 to October 2011. (The Potts method is shown by green squares, the Louvain method by orange circles, and the spectral method by blue triangles.)

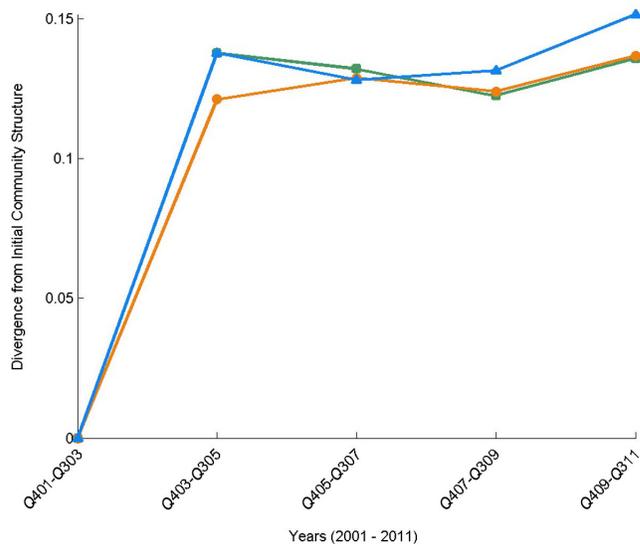


FIG. 17. Temporal trend of VI showing the similarity in community structure for the different algorithms over a two-year sliding window spanning the time frame from October 2001 to October 2011. (The Potts algorithm is shown by green squares, the Louvain algorithm by orange circles, and the spectral algorithm by blue triangles.)

following Eq. (38), our choice of C_{norm} is already discounting (the evolution of) the volatility of the market. Still, we see that $Q_3(\vec{\sigma}^*)$ rises slowly from the period ending in 2003 to the period ending in 2007, implying an increase in the strength of communities, and it then falls by more than 50% by the end of the period ending in 2009. This drop implies a decoherence of the communities throughout that period, quite possibly attributed to the financial crash of 2007–2008. This seems in line with the observation that during periods of financial crisis, markets tend to become more globally correlated, overwhelming the effect of group-level correlations. However, it is interesting that the values of VI have remained quite small and stationary (Fig. 17). This indicates that, despite the fluctuating value of the modularity (i.e., of the relative intracommunity correlations), the composition of the communities has remained very stable over time.

B. Six-month window

We can continue to probe this system at a finer-grained resolution of time periods, to see if the observations made with the two-year window hold up. Again, we plot both $Q_3(\vec{\sigma}^*)$ and VI for the same ten-year period of the S&P and present the results in Figs. 18 and 19, respectively. We can immediately see that the homogeneity of community structure that we see when probing the data using two-year time windows still exists for the most part, but there are some fluctuations in modularity and community composition over the various six-month periods. The graph of $Q_3(\vec{\sigma}^*)$ in Fig. 18 reinforces the observation from Fig. 16 of

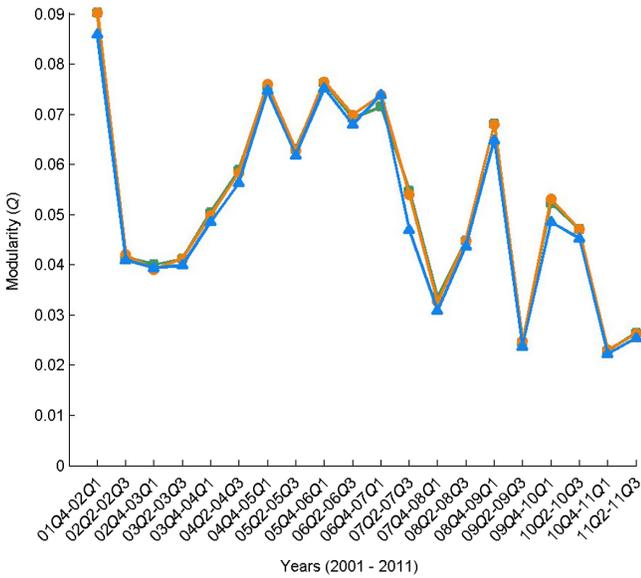


FIG. 18. Temporal trend in the values of $Q_3(\vec{\sigma}^*)$ for the different algorithms over a six-month sliding window spanning the time frame from October 2001 to October 2011. (The Potts algorithm is shown by green squares, the Louvain algorithm by orange circles, and the spectral algorithm by blue triangles.)

a significant drop in modularity around the time of the most recent financial crisis and, more accurately, pinpoints it to the last half of 2007. The VI plot in Fig. 19 indicates again that although the strength of community structure, as measured by $Q_3(\vec{\sigma}^*)$, may have been decreasing, the overall

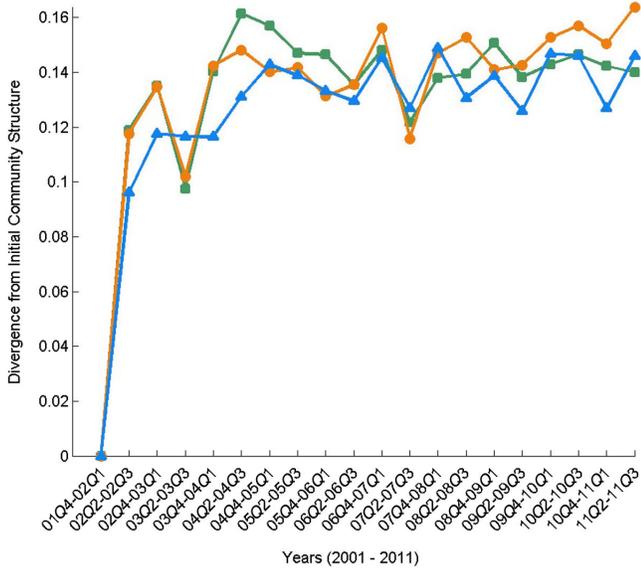


FIG. 19. Temporal trend of VI showing the similarity in community structure for the different algorithms over a six-month sliding window spanning the time frame from October 2001 to October 2011. (The Potts algorithm is shown by green squares, the Louvain algorithm by orange circles, and the spectral algorithm by blue triangles.)

composition of the communities remained relatively constant.

To further examine the coherence and fluctuations in communities across all of the six-month windows, in Fig. 20 we provide a heat map showing the mutual VI between every two pairs of six-month windows. Each square in the matrix is a colored representation of the value of VI between the i th and j th six-month period. Of particular interest, we can see in the lower-right corner of the image (which displays the VI between the most recent time windows) that the communities are slightly more similar than communities generated from the other windows. This indicates that there was less movement of stocks between communities during the most recent couple of years of the past decade. Additionally, these periods are closer to the community structure observed when we measured the entire ten-year period.

As far as explaining this behavior in financial and economical terms, we are again left to hypothesize. Perhaps the observed effect is due to a solidification of communities of stocks caused by the financial collapse, or perhaps it is merely the result of increased accessibility to the markets. With the advent of smartphones, tablets, ease of streaming, and subscribing to news feeds and social networks in conjunction with faster trading systems, and quantitative and high frequency trading, it is conceivable that our increased access to information and the ability to act on it in near real-time has caused a solidifying behavior of the stocks within communities.

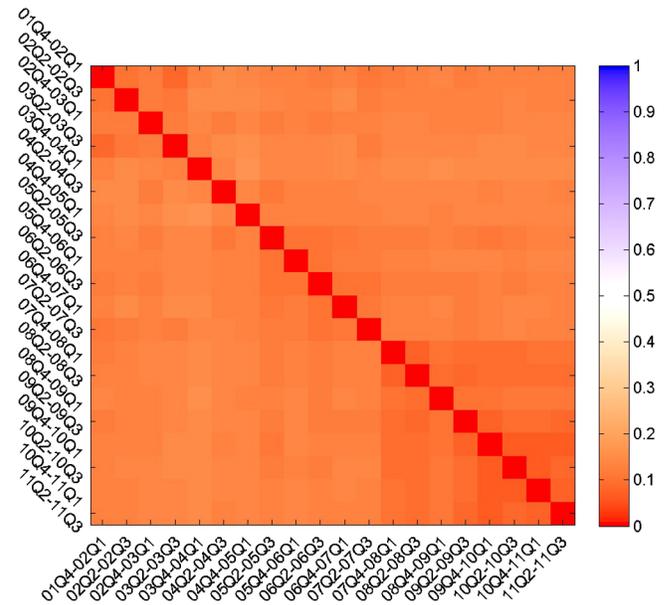


FIG. 20. Heat map showing the value of VI between every pair of six-month time windows, as well as the VI between each window and the total ten-year period (leftmost column and top row). Most notably, there is a slight increase in the similarity of the communities of the last five periods from 2009–2011 (produced using the Louvain algorithm).

C. Temporal coherence of communities: Hard and soft stocks again

We display here one final take on the results obtained from our sliding time window but this time with a stock-centric view, similar to that which we performed for the multifrequency analysis in Sec. VI. In the previous sections, we have alluded to how communities change with time. One question that should be addressed in conjunction with the previous discussion of time scales is then how the composition of a community changes over time. We have already seen from a variety of VI plots that across each of the six-month periods, the sets of communities look slightly different from each other, but what changes are actually taking place? Are there groups of stocks that form tight knit, unwavering cores of communities? Or do they morph fluidly from one to the other, maintaining no coherence over the entire span of ten years?

To address this, we examined the sets of stocks that comprised the communities of each six-month time frame of the S&P over the course of ten years and created a co-occurrence matrix like the one previously shown in Fig. 15, where we calculated the frequency of periods during which any pair of stocks resided in the same community. The resulting heat map is presented in Fig. 21. Again, pairs of stocks that were in the same community all the time are white, and those that were never in the same community are black.

The list of stocks is too long to place on the figure as axis labels, but from observing the raw results, we can make some very interesting observations, which we have tried to summarize by again labeling the graph with GICS industry sectors. We can see that over the course of ten years, the communities do exhibit strong cores which are unwavering

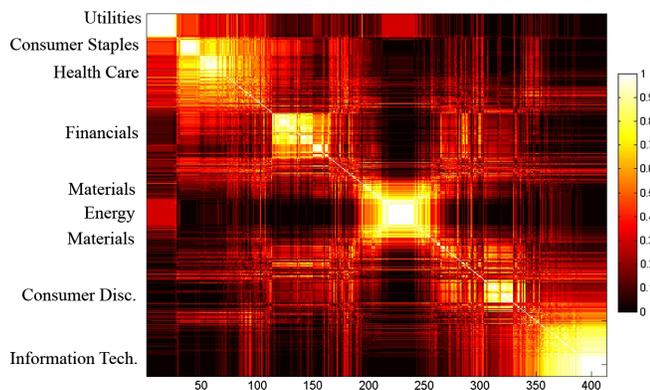


FIG. 21. Coherence of communities over time. The heat map shows the frequency of co-occurrence of different pairs of stocks within the same community over time (white: unit frequency; black: zero frequency). The stocks have been ordered using simulated annealing to position stocks with a high degree of cross-correlations next to each other. To further inform the graph, the GICS sectors have been specified, emphasizing which groupings of stocks tend to associate with a particular sector (produced using the Louvain algorithm).

in their construction and constantly anticorrelated with each other. For example, there exists a set of Core Energy, IT, and Financial stocks that always reside in their own community but never share a community with each other. Groups of Energy, Materials, and Utilities stocks almost always share the same community, but there have been instances when they did not. Finance is broken into a couple of segments of stocks, such as Banks, Real Estate Investment Trusts (REITs), etc., where the smaller groups always trade with each other but are not necessarily aggregated together in a larger community. Similarly, Health Care stocks are fractured in subsets of highly correlated groups of Pharmaceuticals, Services, and Biotech, whose allegiance to the larger industry sectors, such as IT and Consumer Staples, is more fluid. These trends display an interesting overlap with the hierarchical community structure of the S&P discussed earlier in Sec. V F. We also see individual stocks from one top-level industry sector spending most of their time in communities comprised predominantly of a different top-level industry sector, for example Amazon (Consumer Discretionary) spends 90% of the time in the IT group, as does Motorola (Telecommunications).

VIII. CONCLUSIONS

In this paper, we have addressed the challenging problem of the detection of communities of strongly correlated time series, whose importance resides in the possibility of identifying a mesoscopic level of organization in the dynamics of complex systems. While the available techniques to analyze correlation matrices failed to detect such modules, we have shown how the concepts of null models, modularity, and community detection developed in network theory can be appropriately modified in order to successfully cluster matrices of multiple time series. Our redefinitions of the standard methods solve a number of problems encountered when correlation matrices are naively regarded as weighted networks and when ordinary community detection methods are used improperly.

Through the use of various financial markets as examples, we have demonstrated how community detection can be used as a tool to extract specific structural information from time-series data. By surfacing group correlations and trends of the stocks in the S&P 500, the FTSE 100, and the Nikkei 225, we were able to isolate well-defined communities of stocks such that each community exhibited an internal positive correlation between its constituent stocks, where those same stocks exhibited an aggregate residual anticorrelation with the stocks of each of the other communities. While some of these communities showed an association with the more qualitative classification expressed in the GICS industry sector taxonomy, our approach was able to uncover a host of interesting correlations between stocks of different sectors and industry groups, as well as unsuspected residual anticorrelations between stocks of the same sector. As such, our methods

and results show that the observed patterns are irreducible to a standard taxonomy and therefore highlight nontrivial patterns. Moreover, they could prove particularly useful in a number of different fields of finance, such as portfolio optimization and risk management.

It is worth pointing out that our modifications to the Potts, Louvain, and Spectral Optimization algorithms for community detection, although beneficial in and of themselves, act as a proof of concept opening the door to the adaptation of other techniques existing in the field of community detection, allowing, e.g., for overlapping, multiresolution, or hierarchical communities [11,73,74]. Similarly, alternative null models controlling for additional or more sophisticated features of the data can also be developed and incorporated in our approach. The key point is that these models, unlike the naive approach, which has been used so far, should always be consistent with correlation matrices. We hope that our approach will stimulate further research in this direction. Moreover, although we have focused on financial time series as our primary example of real-world data, our general methodology can of course be applied or adapted to any type of time-series data, hopefully yielding equally promising results.

We conclude by noting that, abstractly, the ordinary (network-based) community detection techniques and the (correlation-based) clustering that we have introduced can be thought of as lying at two opposite extremes of a more general problem, in the following sense. The network-based clustering is, in the vast majority of cases, aimed at identifying groups of statically linked objects (as captured by a single temporal snapshot of the network) while disregarding their possibly correlated evolution. By contrast, the correlation-based clustering that we have introduced assumes that the community-defining features are precisely those determining synchronized trends of dynamical activity among nodes, and that the presence (if any) of static dependencies among the latter can be disregarded. One could of course imagine a more general framework where both static linkages and temporal correlations contribute to the definition of communities, possibly overcoming the “functional versus structural” dichotomy such as the one existing in brain network analysis that we mentioned in the Introduction. The present work thus represents one step towards the introduction of a fundamentally more general interpolating formalism.

ACKNOWLEDGMENTS

D.G. acknowledges support from the Dutch Econophysics Foundation (Stichting Econophysics, Leiden, the Netherlands) with funds from beneficiaries of Duyfken Trading Knowledge BV, Amsterdam, the Netherlands. This work was also supported by the EU project MULTIPLEX (Contract No. 317532) and the Netherlands Organization for Scientific Research (NWO/OCW).

APPENDIX: REFINING COMMUNITY DETECTION METHODS

In this appendix, we show that we can successfully reformulate three of the most popular network-based community detection algorithms in order to properly detect communities of correlated time series using the modified modularity function defined in Eq. (36). We stress again that, even if the techniques that we are going to describe can be considered as three different algorithms implementing the same method of modularity maximization, they are often referred to as different “methods” in the literature. In what follows, we will sometimes make use of this somewhat improper terminology. We will also necessarily use a vocabulary that applies more properly to networks than to time series: For instance, a time series will be often denoted as a “node” (or “vertex”) of the “network,” and the correlation between two time series will be denoted as the weight of the “link” (or “edge”) between the corresponding nodes.

1. Modified Potts method

The first of the three methods we have selected is based on the so-called q -state Potts model [42,50]. It represents the system as a q -state spin glass, where each node maintains a spin state σ_i (as given by some attempted partition $\vec{\sigma}$) and the weights of the edges between nodes map to coupling strengths. So any partition of the network is regarded as a spin configuration $\vec{\sigma}$. In this paradigm, the modularity $Q(\vec{\sigma})$ is proportional to the negative energy $-\mathcal{H}(\vec{\sigma})$ of the system. The goal of optimization is then to find the ground state of a spin glass, which corresponds to the maximum value for the modularity. The use of a multistate superparamagnetic model for graph clustering was first introduced by Blatt, Wiseman, and Domany [75] and later revised by Reichardt and Bornholdt [42,50]; it is upon the latter that we base our extension to incorporate multiple time series.

Within the q -state Potts spin-glass model, Reichardt and Bornholdt construct a Hamiltonian by rationalizing a number of energy contributions from the edges between nodes within the same community and nodes in different communities:

$$\begin{aligned} \mathcal{H}(\vec{\sigma}) = & - \sum_{i,j} \overbrace{a_{ij} A_{ij} \delta(\sigma_i, \sigma_j)}^{\text{internal links}} + \sum_{i,j} \overbrace{b_{ij} (1 - A_{ij}) \delta(\sigma_i, \sigma_j)}^{\text{internal non-links}} \\ & + \sum_{i,j} \overbrace{c_{ij} A_{ij} [1 - \delta(\sigma_i, \sigma_j)]}_{\text{external links}} \\ & - \sum_{i,j} \overbrace{d_{ij} (1 - A_{ij}) [1 - \delta(\sigma_i, \sigma_j)]}_{\text{external non-links}}, \end{aligned} \quad (\text{A1})$$

where the contributions from the various types of links can be tuned through the set of coefficients a_{ij} , b_{ij} , c_{ij} , d_{ij} .

Instead of directly maximizing the modularity $Q(\vec{\sigma})$ defined in Eq. (18), Reichardt and Bornholdt minimize the Hamiltonian $\mathcal{H}(\vec{\sigma})$. The latter (under certain conditions and some simplifying assumptions) can be condensed to

$$\mathcal{H}(\vec{\sigma}) = -\sum_{i,j} [A_{ij} - \langle A_{ij} \rangle] \delta(\sigma_i, \sigma_j), \quad (\text{A2})$$

where A_{ij} is the observed value and $\langle A_{ij} \rangle$ is the corresponding null model for that edge. The actual search over spin configurations is done using simulated annealing [76], which is an approximate technique that, in general, returns a different solution each time it is used.

In the same vein, introducing a Hamiltonian corresponding to our correlation-based modularity is equally straightforward; however, we need to ensure that the logic and derivation that were used to develop the original network-based Hamiltonian hold true for a network created from time-series data. For a correlation-based network, we have the situation where every node is connected to every other node, in principle eliminating the energy contributed by nonlinks in Eq. (A1) above. However, as is ordinarily done when applying the Potts model to weighted networks, we can replace the energy contribution of nonlinks with the energy contribution of links whose weight is less than expected, allowing us to immediately introduce our null model. Maintaining the balance between internal and external edges ($a_{ij} = c_{ij}$ and $b_{ij} = d_{ij}$) as was done by Reichardt and Bornholdt in their original derivation of the Hamiltonian, we end up with a variant of Eq. (A2) directly derived from a complete weighted network where A_{ij} and $\langle A_{ij} \rangle$ are replaced by the observed correlation C_{ij} and one of our three null models $\langle C_{ij} \rangle_l$ defined in Sec. IV A, giving

$$\mathcal{H}_l(\vec{\sigma}) = -\sum_{i,j} C_{ij}^{(l)} \delta(\sigma_i, \sigma_j). \quad (\text{A3})$$

Apart from the absence of C_{norm} , the right-hand side of the above expression is the opposite of the right-hand side of Eq. (36). Therefore, our optimization method using the Potts model will attempt to find the lowest value of the Hamiltonian, which will correspond to the highest modularity. For the rest, our algorithm is identical to the procedure described by Reichardt and Bornholdt [42,50].

Therefore, the Potts model is a simple algorithm to adapt to correlation matrices, the reason being that although the modularity of the system is explained using a spin-glass model, the actual optimization process is performed using simulated annealing, which keeps working even if we use our redefinition of modularity as the cost function [76].

2. Modified Louvain method

We now consider a second approach to the problem of modularity optimization. Possibly one of the most successful approaches, the Louvain method [51] (named after the university from which it emerged), is a simple,

greedy, agglomerative algorithm whose strength lies in the fact that it is computationally fast. Unlike the spin-glass model, the Louvain method does not set up a framework for its optimization problem. It simply starts from the definition of modularity specified in Eq. (18) and derives a new, more computationally efficient equation for testing the relative gain in modularity by moving a node from one community to another. It is this equation that allows the Louvain method to perform so well.

The method initially considers all nodes as placed in individual communities and then calculates (sequentially for each node i) the gain of modularity associated with moving node i to the same community where each of its neighbors j belong. The algorithm explores all possible such moves and implements those that give the maximum gain in modularity, and the first iteration stops when no further improvement is possible. Then, a new “renormalized” network is built by merging all nodes within the previously found communities into a single “hypernode,” and the algorithm is iterated again until there are no more possible changes and a maximum of modularity is attained. To do so, the renormalized weight of the link between two hypernodes is defined as the sum of the weight of the links between nodes in the corresponding two communities. Links between nodes of the same community lead to self-loops for the corresponding hypernode.

The key requirement of the Louvain method is that the system can be properly renormalized, i.e., that successive coarse-grainings of the system remain consistent with the meaning of the modularity at the corresponding level of aggregation. In an ordinary network, this is relatively straightforward to show; i.e., a hypernode obtained merging two or more nodes can be legitimately interpreted (from the point of view of the modularity function) as a coarse-grained node with a self-loop to itself and renormalized interactions to all other (hyper)nodes. It is, however, not intuitively obvious whether our modularity defined in Eq. (36) admits an equivalently consistent definition of renormalized time series obtained by merging two or more time series. And even if such a definition exists, one should also understand how to correctly define the renormalized interactions and self-loops. To this end, we recall from Eq. (10) that if X_i and X_j are two standardized time series, then $C_{ij} = \text{Cov}[X_i, X_j]$. We can therefore exploit the fact that the covariance is a bilinear function of its arguments to calculate the following renormalized interactions between two hypernodes (communities) A and B :

$$\begin{aligned} \sum_{i \in A} \sum_{j \in B} C_{ij} &= \sum_{i \in A} \sum_{j \in B} \text{Cov}[X_i, X_j] \\ &= \text{Cov} \left[\sum_{i \in A} X_i, \sum_{j \in B} X_j \right]. \end{aligned} \quad (\text{A4})$$

The above formula shows that, if we define the renormalized time series of community A as

$$\tilde{X}_A \equiv \sum_{i \in A} X_i, \quad (\text{A5})$$

then we can consistently define the renormalized interactions as

$$\tilde{C}_{AB} \equiv \sum_{i \in A} \sum_{j \in B} C_{ij} = \text{Cov}[\tilde{X}_A, \tilde{X}_B] \quad (\text{A6})$$

and the renormalized self-loops as

$$\tilde{C}_{AA} = \text{Cov}[\tilde{X}_A, \tilde{X}_A] = \text{Var}[\tilde{X}_A]. \quad (\text{A7})$$

We therefore find that, for a graph composed of financial time series, renormalized interactions have a correct interpretation in terms of covariances, rather than correlations. They also show that the summation of a group of time series yields something that resembles an index fund of the set of stocks, so the concept of aggregating nodes maintains a strong grounding in reality.

We now have to check whether the modularity function remains consistent with the null model when defined at the level of renormalized nodes. Note that the linearity of the definition of \tilde{C}_{AB} ensures that, given any of our null models defined in Sec. IV A, we can write

$$\langle \tilde{C}_{AB} \rangle_l = \sum_{i \in A} \sum_{j \in B} \langle C_{ij} \rangle_l. \quad (\text{A8})$$

This means that the filtered quantity $C_{ij} - \langle C_{ij} \rangle_l$ can be similarly renormalized as

$$\tilde{C}_{AB}^{(l)} \equiv \sum_{i \in A} \sum_{j \in B} C_{ij}^{(l)} \quad (\text{A9})$$

for each of the three cases in Eq. (37). Now, imagine that in subsequent iterations of the model, the hypernodes are further merged into ‘‘communities of communities.’’ The resulting ‘‘metapartition’’ can be specified by a vector $\vec{\sigma}$ of dimension smaller than (or equal to, if the metapartition is trivial) any of the original vectors $\vec{\sigma}$. Each element $\tilde{\sigma}_A$ denotes the community to which the hypernode A is placed by the metapartition. If $\vec{\sigma}$ denotes the underlying (node-level) partition identified by the metapartition $\vec{\tilde{\sigma}}$ (i.e., $\sigma_i = \tilde{\sigma}_A$ for all $i \in A$), we can define the renormalized modularity

$$\begin{aligned} \tilde{Q}_l(\vec{\tilde{\sigma}}) &\equiv \frac{1}{\tilde{C}_{\text{norm}}} \sum_{A,B} \tilde{C}_{AB}^{(l)} \delta(\tilde{\sigma}_A, \tilde{\sigma}_B) \\ &= \frac{1}{\tilde{C}_{\text{norm}}} \sum_{A,B} \sum_{i \in A} \sum_{j \in B} C_{ij}^{(l)} \delta(\tilde{\sigma}_A, \tilde{\sigma}_B) \\ &= \frac{1}{\tilde{C}_{\text{norm}}} \sum_{i,j} C_{ij}^{(l)} \delta(\sigma_i, \sigma_j) \\ &= Q_l(\vec{\sigma}), \end{aligned} \quad (\text{A10})$$

where, in analogy with Eq. (38), we have defined

$$\tilde{C}_{\text{norm}} \equiv \sum_{A,B} \tilde{C}_{AB} = \sum_{A,B} \sum_{i \in A} \sum_{j \in B} C_{ij} = \sum_{i,j} C_{ij} = C_{\text{norm}}.$$

Equation (A10) coincides with the original modularity defined at the level of individual nodes. This means that the modularity is manifestly invariant under renormalization, implying that we can indeed consistently redefine a coarse-grained modularity at each iteration of the Louvain method.

The second requirement of the Louvain method is the fact that the change in the modularity obtained by adding a previously isolated node to a given preexisting community can be easily calculated. This ensures the computational efficiency of the algorithm. In adapting the model to correlation-based networks, we must start from Eq. (36) and check whether this is still the case and, if so, arrive at a new corresponding expression for the modularity change. We will do so by directly using the invariant modularity defined in Eq. (A10) so that we are sure that the result will hold at any aggregation level. Given the modularity $\tilde{Q}_l(\vec{\tilde{\sigma}})$, we denote the modularity change obtained by adding the (hyper)node I to the community J by $\Delta \tilde{Q}_l^{(I \rightarrow J)}$ and calculate it as the difference between $\tilde{Q}_l(\vec{\tilde{\sigma}}')$ for a (meta)partition $\vec{\tilde{\sigma}}'$ [where I is part of the community J (i.e., $\tilde{\sigma}'_I = J$)] and $\tilde{Q}_l(\vec{\tilde{\sigma}}'')$ for a (meta)partition $\vec{\tilde{\sigma}}''$ [where I is isolated in its own community (i.e., $\tilde{\sigma}''_I \neq J$)]. Since $\tilde{\sigma}'_A = \tilde{\sigma}''_A$ for all $A \neq I$ and $\delta(\tilde{\sigma}'_I, \tilde{\sigma}''_I) = 0$ for all $A \neq I$, we can write this difference as

$$\begin{aligned} \Delta \tilde{Q}_l^{(I \rightarrow J)} &= \tilde{Q}_l(\vec{\tilde{\sigma}}') - \tilde{Q}_l(\vec{\tilde{\sigma}}'') \\ &= \frac{1}{C_{\text{norm}}} \sum_{A,B} \tilde{C}_{AB}^{(l)} [\delta(\tilde{\sigma}'_A, \tilde{\sigma}'_B) - \delta(\tilde{\sigma}''_A, \tilde{\sigma}''_B)] \\ &= \frac{1}{C_{\text{norm}}} \sum_A \tilde{C}_{IA}^{(l)} [\delta(\tilde{\sigma}'_I, \tilde{\sigma}'_A) - \delta(\tilde{\sigma}''_I, \tilde{\sigma}''_A)] \\ &= \frac{1}{C_{\text{norm}}} \sum_{A \in J} \tilde{C}_{IA}^{(l)} \\ &= \frac{\tilde{C}_{IJ}^{(l)}}{C_{\text{norm}}}. \end{aligned} \quad (\text{A11})$$

In other words, the change in modularity obtained from adding a (hyper)node I to a preexisting community J is simply proportional to the renormalized interaction between I and J , i.e., the sum of the (filtered) correlations of all time series within I with all those within J . Note that, in the above formula, the notation $A \in J$ implies $A \neq I$ since I does not (yet) belong to J .

Similarly, it is possible to calculate the change in modularity $-\Delta \tilde{Q}_l^{(I \rightarrow J')}$ obtained when a (hyper)node I belonging to a community J' is disconnected from the latter and placed in its own isolated community. Combining

these two contributions, we can easily calculate the change in modularity

$$-\Delta\tilde{Q}_I^{(I\rightarrow J')} + \Delta\tilde{Q}_I^{(I\rightarrow J)} = \frac{\tilde{C}_{IJ}^{(l)} - \tilde{C}_{IJ'}^{(l)}}{C_{\text{norm}}} \quad (\text{A12})$$

obtained by moving a (hyper)node I from a community J' to a different community J . So our reformulation above also satisfies the second requirement of the Louvain method at all aggregation levels and allows us to define a computationally efficient method to detect communities of time series.

3. Modified spectral method

We now come to the third and final method of optimizing the modularity cost function. Spectral optimization is the process of using matrix eigendecomposition to recursively bisect a network into communities of nodes according to the principle of maximizing the modularity function [52]. The matrix that is the subject of the eigendecomposition is the so-called modularity matrix appearing in Eq. (18) and having entries $B_{ij} = A_{ij} - (k_i k_j / 2m)$. In other words, the modularity matrix \mathbf{B} is the difference between the observed network, represented by the adjacency matrix \mathbf{A} , and the null model $\langle \mathbf{A} \rangle$. In the spectral method, the modularity matrix is eigendecomposed into its constituent eigenvalues and eigenvectors, the intent being to isolate the eigenvector corresponding to the largest eigenvalue and use the signs of the elements of this vector to infer an optimal partition. Specifically, the network is split into two communities, each comprising the nodes corresponding to eigenvector components with the same sign. The process is implemented recursively in each partition (deriving a new modularity matrix for every community), until no further increase in modularity is obtained.

We need to extend this algorithm to accommodate correlation-based networks. In our case, as is clear from Eqs. (36) and (37), the modularity matrix is $\mathbf{C}^{(l)}$, i.e., the filtered matrix defined using one of our three null models. We will therefore adapt the procedure outlined by Newman in the original paper and implement the spectral optimization method by iteratively bisecting the network into two subcommunities (say A and B). Each such bisection can be denoted either by an appropriate partition vector $\vec{\sigma}$ or, equivalently, by a vector \vec{s} having elements $s_i = -1$ if node i belongs to (say) community A and $s_i = +1$ if i belongs to community B . The correspondence between these vectors is given by

$$\delta(\sigma_i, \sigma_j) = \frac{s_i s_j + 1}{2}. \quad (\text{A13})$$

Given a bisection, we can therefore rewrite our unified correlation-based modularity $Q_I(\vec{\sigma})$ defined in Eq. (36) as

$$\begin{aligned} Q_I(\vec{s}) &= \frac{1}{C_{\text{norm}}} \sum_{i,j} C_{ij}^{(l)} \frac{s_i s_j + 1}{2} \\ &= \frac{1}{2C_{\text{norm}}} \sum_{i,j} C_{ij}^{(l)} s_i s_j \\ &\quad + \frac{1}{2C_{\text{norm}}} \sum_{i,j} C_{ij}^{(l)}. \end{aligned} \quad (\text{A14})$$

In Newman's original formulation, the last term sums to zero because the network-based modularity matrix \mathbf{B} has the property that all of its rows sum to zero. However, this is not the case with our correlation-based modularity matrix $\mathbf{C}^{(l)}$ defined in Eq. (37). So we retain the second term and, defining $C_{\text{tot}}^{(l)} \equiv \sum_{i,j} C_{ij}^{(l)}$, rewrite our modularity in matrix form as

$$Q_I(\vec{s}) = \frac{\langle s | \mathbf{C}^{(l)} | s \rangle}{2C_{\text{norm}}} + \frac{C_{\text{tot}}^{(l)}}{2C_{\text{norm}}}. \quad (\text{A15})$$

The vector \vec{s} maximizing $Q_I(\vec{s})$ is easily found as the vector matching the signs of the components of the eigenvector of $\mathbf{C}^{(l)}$ corresponding to the largest eigenvalue. Clearly, neither C_{norm} nor $C_{\text{tot}}^{(l)}$ has an effect on the result, making the original procedure of the spectral algorithm consistent with our reformulation.

After the initial bisection, we need to calculate the potential modularity change ΔQ_I obtained by further subdividing the communities yielded in the previous step. Let us consider the case where one community (say, A) among the ones obtained thus far in the algorithm is further subdivided into two new communities (say, A_1 and A_2). If \vec{s} is a vector (restricted to the vertices in A only) denoting the bisection of A into A_1 and A_2 , then the modularity change associated with such bisection reads

$$\begin{aligned} \Delta Q_I^{(A_1|A_2)} &= \frac{1}{C_{\text{norm}}} \left[\sum_{i,j \in A_1} C_{ij}^{(l)} + \sum_{i,j \in A_2} C_{ij}^{(l)} - \sum_{i,j \in A} C_{ij}^{(l)} \right] \\ &= \frac{1}{C_{\text{norm}}} \left[\sum_{i,j \in A} C_{ij}^{(l)} \frac{s_i s_j + 1}{2} - \sum_{i,j \in A} C_{ij}^{(l)} \right] \\ &= \frac{1}{2C_{\text{norm}}} \left[\sum_{i,j \in A} C_{ij}^{(l)} s_i s_j - \sum_{i,j \in A} C_{ij}^{(l)} \right] \\ &= \frac{\langle s | \mathbf{C}_A^{(l)} | s \rangle}{2C_{\text{norm}}} - \frac{\tilde{C}_{AA}^{(l)}}{2C_{\text{norm}}}, \end{aligned} \quad (\text{A16})$$

where $\mathbf{C}_A^{(l)}$ represents the submatrix of $\mathbf{C}^{(l)}$ restricted to the subset of nodes within community A , and the notation $\tilde{C}_{AA}^{(l)}$ is borrowed from Eq. (A9). As with the initial bisection, \vec{s} is chosen to maximize $\Delta Q_I^{(A_1|A_2)}$ by selecting its elements to

match the sign of the eigenvector corresponding to the largest eigenvalue of the matrix $\mathbf{C}_A^{(l)}$.

As for the original algorithm, our modified spectral method proceeds by iterating the above procedure until no further bisection can make the modularity increase.

-
- [1] P. Lyman and H. R. Varian, *How Much Information?*, <http://www2.sims.berkeley.edu/research/projects/how-much-info-2003>.
- [2] R. Neuman, Y. J. Park, and E. Panek, *Tracking the Flow of Information into the Home: An Empirical Assessment of the Digital Revolution in the United States, 1960–2005*, *Int. J. Comm.* **6**, 1022 (2012).
- [3] H. Park and K. Friston, *Structural and Functional Brain Networks: From Connections to Cognition*, *Science* **342**, 1238411 (2013).
- [4] S. Sinha, A. Chatterjee, A. Chakraborti, and B. K. Chakrabarti, *Econophysics* (Wiley-VCH, Weinheim, 2011).
- [5] J. P. Bouchaud and M. Potters, *Theory of Financial Risk and Derivative Pricing*, 2nd ed. (Cambridge University Press, Cambridge, England, 2003).
- [6] A. Utsugi, K. Ino, and M. Oshikawa, *Random Matrix Theory Analysis of Cross Correlations in Financial Markets*, *Phys. Rev. E* **70**, 026110 (2004).
- [7] M. Potters, J. P. Bouchaud, and L. Laloux, *Financial Applications of Random Matrix Theory: Old Laces and New Pieces*, *Acta Phys. Pol. B* **36**, 2767 (2005).
- [8] V. Plerou, P. Gopikrishnan, B. Rosenow, L. A. N. Amaral, T. Guhr, and H. E. Stanley, *Random Matrix Approach to Cross Correlations in Financial Data*, *Phys. Rev. E* **65**, 066126 (2002).
- [9] R. N. Mantegna and H. E. Stanley, *Introduction to Econophysics: Correlations and Complexity in Finance* (Cambridge University Press, Cambridge, England, 1999).
- [10] One might think of community detection itself as a geometric method, but this is not what we mean here. The geometric methods we are referring to consist in embedding techniques that reduce the complexity of the original system by projecting the latter into some metric space of low dimensionality.
- [11] S. Fortunato, *Community Detection in Graphs*, *Phys. Rep.* **486**, 75 (2010).
- [12] M. E. J. Newman, *Networks, An Introduction*, (Oxford University Press, Oxford, 2010).
- [13] T. Heimo, J. M. Kumpula, K. Kaski, and J. Saramäki, *Detecting Modules in Dense Weighted Networks with the Potts Method*, *J. Stat. Mech.* **2008**, P08007 (2008).
- [14] D. J. Fenn, M. A. Porter, P. J. Mucha, M. McDonald, S. Williams, N. F. Johnson, and N. S. Jones, *Dynamical Clustering of Exchange Rates*, *Quant. Finance* **12**, 1493 (2012).
- [15] T. Isogai, *Clustering of Japanese Stock Returns by Recursive Modularity Optimization for Efficient Portfolio Diversification*, *J. Complex Networks* **2**, 557 (2014).
- [16] C. Piccardi, L. Calatroni, and F. Bertoni, *Clustering Financial Time Series by Network Community Analysis*, *Int. J. Mod. Phys. C* **22**, 35 (2011).
- [17] E. P. Wigner, *Characteristic Vectors of Bordered Matrices with Infinite Dimensions*, *Ann. Math.* **62**, 548 (1955).
- [18] M. L. Mehta, *Random Matrices* (Elsevier, 2004).
- [19] J. P. Onnela, K. Kaski, and J. Kertsz, *Clustering and Information in Correlation Based Financial Networks*, *Eur. Phys. J. B* **38**, 353 (2004).
- [20] T. Heimo, K. Kimmo, and J. Saramaki, *Maximal Spanning Trees, Asset Graphs and Random Matrix Denoising in the Analysis of Dynamics of Financial Networks*, *Physica A* **388**, 145 (2009).
- [21] R. Rammal, G. Toulouse, and M. A. Virasoro, *Ultrametricity for Physicists*, *Rev. Mod. Phys.* **58**, 765 (1986).
- [22] T. Aste, T. Di Matteo, and S. T. Hyde, *Complex Networks on Hyperbolic Surfaces*, *Physica A* **346**, 20 (2005).
- [23] M. Tumminello, T. Aste, T. Di Matteo, and R. N. Mantegna, *A Tool for Filtering Information in Complex Systems*, *Proc. Natl. Acad. Sci. U.S.A.* **102**, 10421 (2005).
- [24] W. M. Song, T. Di Matteo, and T. Aste, *Nested Hierarchy in Planar Graphs*, *Discrete Appl. Math.* **159**, 2135 (2011).
- [25] W. M. Song, T. Di Matteo, and T. Aste, *Hierarchical Information Clustering by Means of Topologically Embedded Graphs*, *PLoS One* **7**, e31929 (2012).
- [26] R. Morales, T. Di Matteo, and T. Aste, *Dependency Structure and Scaling Properties of Financial Time Series Are Related*, *Sci. Rep.* **4**, 4589 (2014).
- [27] V. Plerou, P. Gopikrishnan, B. Rosenow, L. A. N. Amaral, and H. E. Stanley, *Universal and Nonuniversal Properties of Cross Correlations in Financial Time Series*, *Phys. Rev. Lett.* **83**, 1471 (1999).
- [28] L. Laloux, P. Cizeau, J. P. Bouchaud, and M. Potters, *Noise Dressing of Financial Correlation Matrices*, *Phys. Rev. Lett.* **83**, 1467 (1999).
- [29] M. Tumminello, F. Lillo, and R. N. Mantegna, *Spectral Properties of Correlation Matrices for Some Hierarchically Nested Factor Models*, *AIP Conf. Proc.* **965**, 300 (2007).
- [30] M. Tumminello, F. Lillo, and R. N. Mantegna, *Hierarchically Nested Factor Model from Multivariate Data*, *Europhys. Lett.* **78**, 30006 (2007).
- [31] N. Higham, *Computing the Nearest Correlation Matrix—A Problem from Finance*, *IMA J. Numer. Anal.* **22**, 329 (2002).
- [32] R. Rebonato and P. Jackel, *The Most General Methodology for Creating a Valid Correlation Matrix for Risk Management and Option Pricing Purposes*, *J. Risk* **2**, 17 (1999).
- [33] J. Simonian, *The Most Simple Methodology to Create a Valid Correlation Matrix for Risk Management and Option Pricing Purposes*, *Appl. Econ. Lett.* **17**, 1767 (2010).
- [34] O. Ledoit and M. Wolf, *Improved Estimation of the Covariance Matrix of Stock Returns with an Application to Portfolio Selection*, *J. Empirical Finance* **10**, 603 (2003).
- [35] M. Tumminello, F. Lillo, and R. N. Mantegna, *Kullback-Leibler Distance as a Measure of the Information Filtered from Multivariate Data*, *Phys. Rev. E* **76**, 031123 (2007).
- [36] M. E. J. Newman and M. Girvan, *Finding and Evaluating Community Structure in Networks*, *Phys. Rev. E* **69**, 026113 (2004).
- [37] M. A. Porter, J. P. Onnela, and P. J. Mucha, *Communities in Networks*, *Not. Am. Math. Soc.* **56(9)**, 1082 (2009).

- [38] D. Garlaschelli and M. I. Loffredo, *Maximum Likelihood: Extracting Unbiased Information from Complex Networks*, *Phys. Rev. E* **78**, 015101 (2008).
- [39] T. Squartini and D. Garlaschelli, *Analytical Maximum-Likelihood Method to Detect Patterns in Real Networks*, *New J. Phys.* **13**, 083001 (2011).
- [40] D. Garlaschelli and M. I. Loffredo, *Generalized Bose-Fermi Statistics and Structural Correlations in Weighted Networks*, *Phys. Rev. Lett.* **102**, 038701 (2009).
- [41] S. Fortunato and M. Barthelemy, *Resolution Limit in Community Detection*, *Proc. Natl. Acad. Sci. USA* **104**, 36 (2007).
- [42] J. Reichardt and S. Bornholdt, *Detecting Fuzzy Community Structures in Complex Networks with a Potts Model*, *Phys. Rev. Lett.* **93**, 218701 (2004).
- [43] A. Arenas, A. Fernández, and S. Gómez, *Analysis of the Structure of Complex Networks at Different Resolution Levels*, *New J. Phys.* **10**, 053039 (2008).
- [44] R. Lambiotte, *Multi-Scale Modularity in Complex Networks*, in *Proceedings of the 8th International Symposium on Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks (WiOpt)*, (IEEE, 2010), p. 546.
- [45] This is true as long as the degrees are *graphic*, i.e., realized by at least one network. This property is always ensured here since the values of the degrees used in the null model are the observed ones and are therefore realized by at least the empirical network.
- [46] S. Gómez, P. Jensen, and A. Arenas, *Analysis of Community Structure in Networks of Correlated Data*, *Phys. Rev. E* **80**, 016114 (2009).
- [47] For homogeneously sized communities, the problem of nonunit diagonal entries can be avoided if we require $i \neq j$ in the summation defining Q . However, for heterogeneously sized community, the bias of the naive approach cannot be eliminated.
- [48] <http://www.lorenz.leidenuniv.nl/~garlaschelli/>.
- [49] M. MacMahon, <http://www.mathworks.com/matlabcentral/fileexchange/49011>.
- [50] J. Reichardt and S. Bornholdt, *Statistical Mechanics of Community Detection*, *Phys. Rev. E* **74**, 016110 (2006).
- [51] V. D. Blondel, J. Guillaume, R. Lambiotte, and E. Lefebvre, *Fast Unfolding of Communities in Large Networks*, *J. Stat. Mech.* 2008, P10008 (2008).
- [52] M. E. J. Newman, *Modularity and Community Structure in Networks*, *Proc. Natl. Acad. Sci. USA* **103**, 8577 (2006).
- [53] E. F. Fama and K. R. French, *Common Risk Factors in the Returns on Stocks and Bonds*, *J. Financ. Econ.* **33**, 3 (1993).
- [54] E. F. Fama and K. R. French, *Multifactor Explanations of Asset Pricing Anomalies*, *J. Finance* **51**, 55 (1996).
- [55] W. F. Sharpe, *Asset Allocation: Management Style and Performance Measurement*, *J. Portfolio Management* **18**, 7 (1992).
- [56] M. Meila, *Comparing Clusterings, An Information Based Distance*, *J. Multivariate Anal.* **98**, 873 (2007).
- [57] M. Meila, *Comparing Clusterings by the Variation of Information*, *Learning Theory Kernel Machines* **2777**, 173 (2003).
- [58] Standard & Poor's developed what is known as the Global Industry Classification Standard as an industry taxonomy, for use by the financial sector. All stocks in the three indexes we work with are classified using this single taxonomy.
- [59] <http://www.standardandpoors.com/products-services/GICS/en/us>.
- [60] D. H. Kim and H. Jeong, *Systematic Analysis of Group Identification in Stock Markets*, *Phys. Rev. E* **72**, 046133 (2005).
- [61] R. K. Pan and S. Sinha, *Collective Behavior of Stock Price Movements in an Emerging Market*, *Phys. Rev. E* **76**, 046116 (2007).
- [62] R. A. Fisher, *Frequency Distribution of the Values of the Correlation Coefficients in Samples from an Indefinitely Large Population*, *Biometrika* **10**, 507 (1915).
- [63] M. Bastian, S. Heymann, and M. Jacomy, *Gephi: An Open Source Software for Exploring and Manipulating Networks*, International AAAI Conference on Weblogs and Social Media, <http://www.aaai.org/ocs/index.php/ICWSM/09/paper/viewFile/154Forum/1009>.
- [64] T. M. J. Fruchterman and E. M. Reingold, *Graph Drawing by Force-Directed Placement*, *Software: Practice and Experience*, **21**, 1129 (1991).
- [65] S. J. Kon, *Models of Stock Returns—A comparison*, *J. Finance* **39**, 147 (1984).
- [66] R. Cont, *Empirical Properties of Asset Returns; Stylized Facts and Statistical Issues*, *Quant. Finance* **1**, 223 (2001).
- [67] C. Curme, M. Tumminello, R. N. Mantegna, H. E. Stanley, and D. Y. Kenett, *Emergence of Statistically Validated Financial Intraday Lead-Lag Relationships*, *arXiv: 1401.0462v1*.
- [68] H. M. Markowitz, *Portfolio Selection*, *J. Finance* **7**, 77 (1952).
- [69] H. M. Markowitz, *Portfolio Selection: Efficient Diversification of Investments* (John Wiley & Sons, 1959).
- [70] E. J. Elton and M. J. Gruber, *Modern Portfolio Theory and Investment Analysis* (Wiley, 1995).
- [71] The GICS hierarchy follows the form of “Sector” “Industry Group” “Industry,” and so the examples in this section will follow the same form, e.g., “Information Technology” “Software & Services” “Software.” However, since our discussion will predominantly focus on similarities at the Industry level, for brevity we will often omit the full GICS label. In such cases, we will delineate this using italics, e.g., *Software* in the example above.
- [72] B. H. Good, Y. de Montjoye, and A. Clauset, *Performance of Modularity Maximization in Practical Contexts*, *Phys. Rev. E* **81**, 046106 (2010).
- [73] P. Ronhovde and Z. Nussinov, *Multiresolution Community Detection for Megascale Networks by Information-Based Replica Correlations*, *Phys. Rev. E* **80**, 016109 (2009).
- [74] P. Ronhovde, S. Chakrabarty, D. Hu, M. Sahu, K. K. Sahu, K. F. Kelton, N. A. Mauro, and Z. Nussinov, *Detection of Hidden Structures for Arbitrary Scales in Complex Physical Systems*, *Sci. Rep.* **2**, 329 (2012).
- [75] M. Blatt, S. Wiseman, and E. Domany, *Superparamagnetic Clustering of Data*, *Phys. Rev. Lett.* **76**, 3251 (1996).
- [76] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, *Optimization by Simulated Annealing*, *Science* **220**, 671 (1983).