Alignment and integration of complex networks by hypergraph-based spectral clustering

Tom Michoel[*]

Freiburg Institute for Advanced Studies (FRIAS), University of Freiburg, Albertstrasse 19, D-79104 Freiburg, Germany and The Roslin Institute, The University of Edinburgh, Easter Bush, Midlothian, EH25 9RG, Scotland, United Kingdom

Bruno Nachtergaele[†]

Department of Mathematics, University of California Davis, One Shields Avenue, Davis, California 95616-8366, USA

(Received 16 May 2012; published 26 November 2012)

Complex networks possess a rich, multiscale structure reflecting the dynamical and functional organization of the systems they model. Often there is a need to analyze multiple networks simultaneously, to model a system by more than one type of interaction, or to go beyond simple pairwise interactions, but currently there is a lack of theoretical and computational methods to address these problems. Here we introduce a framework for clustering and community detection in such systems using hypergraph representations. Our main result is a generalization of the Perron-Frobenius theorem from which we derive spectral clustering algorithms for directed and undirected hypergraphs. We illustrate our approach with applications for local and global alignment of protein-protein interaction networks between multiple species, for tripartite community detection in folksonomies, and for detecting clusters of overlapping regulatory pathways in directed networks.

DOI: 10.1103/PhysRevE.86.056111 PACS number(s): 89.75.Fb, 89.65.−s, 87.18.Vf

I. INTRODUCTION

Complex networks in nature and society represent interactions between entities in inhomogeneous systems and understanding their structure and function has been the focus of much research. On the macroscopic scale, complex networks are characterized by, among others, a degree distribution, characteristic path length, and clustering coefficient, which are markedly different from those of regular lattices or uniformly distributed Erdős-Rényi random graphs [1,2], while on the microscopic scale, they contain network motifs, i.e., small subgraphs occurring significantly more often than expected by chance [3]. The intermediate level usually exhibits the presence of communities or modules, i.e., sets of nodes with a significantly higher than expected density of links between them, with typical examples being friendship circles in social networks, websites devoted to similar topics in the World Wide Web, or protein complexes in protein interaction networks [4–7].

However, the limitations of modeling a complex system by a network with a single type of pairwise interaction are becoming more and more clear. Folksonomies, i.e., online social communities where users apply tags to annotate resources such as images or scientific articles, have a tripartite structure with three types of interactions [8,9]. In biology, cellular systems are characterized by different types of networks which represent different physical interaction mechanisms operating on different time scales, intertwined with each other through extensive feedforward and feedback loops [10,11]. To understand how evolutionary dynamics shapes molecular interaction networks, we need to compare them between multiple species with nontrivial many-to-many relations between their respective node sets [12]. In order to move beyond simple networks of pairwise interactions to model these and other systems, one suggestion has been to use hypergraphs, where edges are arbitrarily sized subsets of nodes. Although a number of studies have generalized various concepts, from graph theory to hypergraphs [8,9,13–16], a rigorous mathematical foundation and general-purpose algorithm for clustering and community detection in hypergraphs is still lacking.

Here we present a framework for spectral clustering in hypergraphs, which is mathematically sound and algorithmically efficient. It is based on a generalization of the Perron-Frobenius theorem, which allows one to define and compute a dominant eigenvector for hypergraphs and use its values for optimally partitioning the hypergraph's vertex set, similar to the operation of standard spectral clustering algorithms in ordinary graphs [17]. We demonstrate the validity of our approach through practical applications in the analysis of real-world networks. In particular, we address the following problems. First, if two networks are defined on separate node sets with a many-to-many mapping between them (for instance, protein-protein interaction networks in different species), it is a natural question to find matching communities in the two networks. This is the so-called *network alignment* problem [12]. We show that this problem can be solved by finding clusters in a hypergraph where each hyperedge consists of two matching edges, with one from each network (see Sec. VIII A). Second, if multiple networks are defined on the same node set (i.e., together they form an edge-colored graph), there often exist functionally meaningful, higher-order relations between the different edge types (for instance, tripartite relations in folksonomies [8,9] or network motifs in biological networks [10,11]). Finding communities or modules with respect to these higher-order relations is what we call the *network integration* problem. Here we show that any higher-order edge relation between different networks defines a subgraph pattern in the corresponding edge-colored graph and that all instances of this pattern form a hypergraph. Hypergraph-based clustering can then be applied to identify modules in such edge-colored graphs (see Secs. VIII B and VIII C).

[*] tom.michoel@roslin.ed.ac.uk

[†] bxn@math.ucdavis.edu

II. GRAPHS AND HYPERGRAPHS

A graph $\mathcal{G}$ is defined as a pair $(\mathcal{V},\mathcal{E})$ of vertices $\mathcal{V}$ and edges (pairs of vertices) $\mathcal{E}$, which may or may not be directed. In a weighted graph, a number is assigned to each edge which may represent, e.g., the cost, length, or reliability of an edge. A hypergraph is a generalization of a graph where an edge, called hyperedge in this case, can connect any number of vertices, i.e., $\mathcal{E}$ is a set of arbitrarily sized subsets of $\mathcal{V}$. A particular class of hypergraphs are so-called k-uniform hypergraphs where each hyperedge has the same cardinality k. Algebraically, a graph can be represented by an adjacency matrix A of dimension $N \times N$, with N the number of vertices, such that $A_{ij} = 1$ if $\{i,j\} \in \mathcal{E}$, and 0 otherwise. For undirected graphs, A is a symmetric matrix, and for weighted graphs, A_{ij} is defined to be the weight of the edge $\{i,j\}$. For k-uniform hypergraphs, the notion of adjacency matrix can be generalized to an adjacency multiarray or tensor T, with $T_{i_1,\ldots,i_k} = 1$ if $\{i_1,\ldots,i_k\} \in \mathcal{E}$, and 0 otherwise. For a general hypergraph, we define a function w on the set of subsets of $\mathcal{V}$ such that $w(E) = 1$ for $E \in \mathcal{E}$, and 0 otherwise. In general, we allow weighted hypergraphs where w can be any non-negative function.

A path between two vertices i and j in a hypergraph is defined as a sequence of vertices $i = i_1, i_2, \ldots, i_{k+1} = j$ and edges $E_1,\ldots,E_k$ such that for all m, $\{i_m, i_{m+1}\} \subset E_m$. A hypergraph is called *connected* if there exists a path between any pair of vertices. A stronger constraint on the structure of a hypergraph is that of *irreducibility*. A hypergraph is said to be reducible if there exists a proper vertex subset $I \subset \mathcal{V}$ such that for any $i \in I$ and $j_1,\ldots,j_m \notin I$, $w(\{i,j_1,\ldots,j_m\}) = 0$, and is said to be irreducible if it is not reducible. For ordinary graphs, connectedness and irreducibility are equivalent, but for hypergraphs this is not the case. An irreducible hypergraph is clearly connected, but the opposite is not always true. Indeed, if there exists a subset of vertices I such that paths crossing from $i \in I$ to $j \notin I$ can always be chosen to do so through an edge of the form $\{i_1,\ldots,i_k,j_1,\ldots,j_m\}$, with $k \geqslant 2$, $i_1,\ldots,i_k \in I$, and $j_1,\ldots,j_m \notin I$, then we can set $w(\{i,j_1,\ldots,j_m\}) = 0$ for all $i \in I$ and $j_1,\ldots,j_m \notin I$, thereby making the hypergraph reducible, without breaking its connectivity.

Directed hypergraphs can be defined in many ways. For instance, for k-uniform hypergraphs, we can impose any form of permutation symmetry, or lack thereof, between some or all of the k dimensions in each edge. In this paper, we will only consider the case where each edge E can be written as a pair (S,T), where $S \subset \mathcal{V}$ is called the "source" vertex set and $T \subset \mathcal{V}$ is called the "target" vertex set, with weight function $w(S,T)$. Underlying a directed hypergraph, there is always an undirected hypergraph with edges $E = S \cup T$ for every directed edge (S,T). As is the case for ordinary directed graphs, a stronger notion of connectivity is usually needed than simple connectivity of this undirected hypergraph. We defer the somewhat technical definition of strong connectivity of directed hypergraphs to Appendix A.

III. DOMINANT EIGENVECTORS AND SPECTRAL GRAPH CLUSTERING

Although countless measures have been designed to define clusters in a graph [5–7], perhaps the simplest definition is that a cluster is a subset of vertices with a high number of edges between them, relative to its size. Mathematically, for a graph with adjacency matrix A, the edge-to-node ratio of a subset $X \subset \mathcal{V}$ can be written as

$$\mathcal{S}(X) = \frac{\sum_{i,j\in X} A_{ij}}{|X|},$$

where $|X|$ denotes the number of elements in X. The number of subsets of a set with N elements grows exponentially in N and hence finding the subset with a maximal edge-to-node ratio by exhaustive enumeration is computationally infeasible for large graphs. However, if we denote by u_X the unit vector in $\mathbb{R}^N$ which has $u_{X,i} = |X|^{-1/2}$ for $i \in X$, and 0 otherwise, we can write $\mathcal{S}$ as a scalar product and obtain the simple upper bound:

$$\mathcal{S}(X) = \langle u_X, A u_X\rangle \leqslant \max_{x\in\mathbb{R}^N, x\neq 0} \frac{\langle x, Ax\rangle}{\|x\|^2} = \lambda_{\max}, \qquad (1)$$

where $\langle x,y\rangle = \sum_i x_i y_i$ is the standard inner product on $\mathbb{R}^N$, $\|x\| = \sqrt{\langle x,x\rangle}$ is the length of x, and $\lambda_{\max}$ is the largest eigenvalue of A. By the Perron-Frobenius theorem [18], if the graph is irreducible, then the dominant eigenvector x, which satisfies $\lambda_{\max} x = Ax$, is unique, strictly positive ($x_i > 0$ for all i), and solves the variational problem on the right-hand side of Eq. (1).

Hence, to find an approximate maximizer X of $\mathcal{S}$, we can take the set X for which u_X is as close as possible to the dominant eigenvector x, which is similar to what is done in other spectral clustering algorithms based on the Laplacian or modularity matrices [17], i.e., define

$$\tilde{X} = \underset{X\subset\mathcal{V}}{\operatorname{argmax}} \langle u_X, x\rangle = \underset{X\subset\mathcal{V}}{\operatorname{argmax}} \frac{1}{|X|^{1/2}} \sum_{i\in X} x_i.$$

Since $x > 0$, $\tilde{X}$ is of the form $X_c = \{i : x_i > c\}$ for some threshold value c. Instead of $\tilde{X}$, we therefore choose the solution of the restricted variational problem

$$X_{\max} = \underset{c>0}{\operatorname{argmax}}\, \mathcal{S}(X_c) \qquad (2)$$

as an approximate maximizer. Solving Eq. (2) is linear in the number of vertices, since we only need to consider the values c equal to the entries of x. Moreover, $\mathcal{S}(X_{\max}) \geqslant \mathcal{S}(\tilde{X})$, and hence $X_{\max}$ is a better approximation to the true maximizer of $\mathcal{S}$ than $\tilde{X}$.

Thus we obtain a numerically highly efficient spectral graph clustering algorithm:

(1) Calculate the dominant eigenvector x using, for instance, a power method [19].

(2) Find the cluster $X_{\max}$ which solves the restricted variational problem in Eq. (2).

(3) Store $X_{\max}$, remove all edges between nodes in $X_{\max}$ from the edge set $\mathcal{E}$, and repeat the procedure until no more edges remain.

This result of this algorithm is a partition of the edges of the input graph. Edge clustering algorithms have recently gained popularity, as they allow for overlapping communities where nodes may belong to more than one community [20,21].

This procedure generalizes immediately to directed or bipartite graphs. In this case, a cluster consists of a source set X and target set Y with edge-to-node ratio

$$\mathcal{S}(X,Y) = \frac{\sum_{i\in X, j\in Y} A_{ij}}{\sqrt{|X|\cdot|Y|}}.$$

The dominant eigenvector is replaced by the dominant left and right singular vectors x and y corresponding to the largest singular value of A, which are again unique and strictly positive [18]. $X_{\max}$ and $Y_{\max}$ are found by maximizing $\mathcal{S}(X,Y)$ over sets obtained by thresholding on the entries of x and y.

IV. PERRON-FROBENIUS THEOREM FOR HYPERGRAPHS

Our aim is to generalize the previous graph spectral clustering algorithm to arbitrary hypergraphs. For this purpose, we first need a generalization of the Perron-Frobenius theorem. Let $\mathcal{H} = (\mathcal{V},\mathcal{E})$ be an undirected hypergraph on N vertices. Define, for $x \in \mathbb{R}^N$ and $p \geqslant 1$,

$$\mathcal{R}_p(x) = \sum_{E\in\mathcal{E}} w(E) \prod_{i\in E} \left(\frac{|x_i|}{\|x\|_p}\right)^{\frac{1}{|E|}}, \tag{3}$$

where $w(E)$ is the non-negative weight of edge E and $\|x\|_p = (\sum_i |x_i|^p)^{1/p}$ is the p norm of x. We have the following key result:

Theorem. $\mathcal{R}_p$ attains its maximum on the set of unit vectors $\mathbb{S}_p^N = \{u \in \mathbb{R}^N : \|u\|_p = 1\}$. If $\mathcal{H}$ is connected, there is a unique maximizer $x \in \mathbb{S}_p^N$ which is strictly positive and satisfies the Euler-Lagrange equations

$$\lambda_p x_i^p = \sum_{\{E\in\mathcal{E}: i\in E\}} \frac{w(E)}{|E|} \left(\prod_{j\in E} x_j\right)^{\frac{1}{|E|}}, \tag{4}$$

subject to the constraint $\|x\|_p = 1$ and with $\lambda_p = \mathcal{R}_p(x)$. By analogy with the matrix case, we call x the dominant eigenvector of $\mathcal{H}$.

For clarity, we first prove this theorem in the simpler case when $\mathcal{H}$ is irreducible. The proof of the general case is given in Appendix B.

Proof. Existence of a maximizer on $\mathbb{S}_p^N$ follows from Weierstrass's theorem [18]. Clearly, since $\mathcal{R}_p(x) = \mathcal{R}_p(|x|)$, we can always choose a maximizer x to have non-negative entries. Hence we can find x as a stationary point of the Lagrangian

$$\mathcal{L}(x) = \sum_{E\in\mathcal{E}} w(E) \left(\prod_{i\in E} |x_i|\right)^{\frac{1}{|E|}} - \frac{\lambda}{p}\left(\|x\|_p^p - 1\right),$$

giving rise (for non-negative x) to the Euler-Lagrange equations

$$\lambda x_i^{p-1} = \sum_{\{E\in\mathcal{E}: i\in E\}} \frac{w(E)}{|E|} \left(\prod_{j\in E, j\neq i} x_j\right)^{\frac{1}{|E|}} x_i^{\frac{1}{|E|}-1}. \tag{5}$$

Let $I = \{i \in \mathcal{V} : x_i = 0\}$ and $i \in I$. Assume there exists an edge $E = \{i, j_1, \ldots, j_m\}$ with $j_1, \ldots, j_m \notin I$. Then the left-hand side of Eq. (5) is 0, while the right-hand side is ∞. Hence such an edge cannot exist, but this contradicts the assumption of irreducibility of $\mathcal{H}$. It follows that $I = \emptyset$ or $x > 0$. Multiplying both sides of Eq. (5) by x_i, we obtain Eq. (4). Summing both sides in Eq. (4) over i gives $\lambda_p = \mathcal{R}_p(x) = \max_{x'} \mathcal{R}_p(x')$.

Next assume $y > 0$ is another maximizer of $\mathcal{R}_p$. Denote $c = \min_i(x_i/y_i)$, $u = cy$, and $z = x - u \geqslant 0$. Since $\|x\|_p = \|y\|_p = 1$, we have $c < 1$ and $c^p \leqslant c$ for $p \geqslant 1$. Denote $I = \{i \in \mathcal{V} : z_i = 0\}$. For any $i \in I$, by the Euler-Lagrange equations,

$$\begin{aligned} 0 &= \lambda_p\left(x_i^p - c^p y_i^p\right) \\ &\geqslant \sum_{\{E\in\mathcal{E}: i\in E\}} w(E) \left[\left(\prod_{j\in E} x_j\right)^{\frac{1}{|E|}} - \left(\prod_{j\in E} u_j\right)^{\frac{1}{|E|}}\right]. \end{aligned}$$

Since each term in the last sum is non-negative, they must all be zero. Hence for any $j_1, \ldots, j_k \notin I$, if $\{i, j_1, \ldots, j_k\} \in \mathcal{E}$, then

$$\begin{aligned} 0 &= \prod_{m=1}^{k} x_{j_m} - \prod_{m=1}^{k} u_{j_m} \\ &= \sum_{m=1}^{k} \left(\prod_{n=1}^{m-1} u_{j_n}\right)\left(x_{j_m} - u_{j_m}\right)\left(\prod_{n=m+1}^{k} x_{j_n}\right). \end{aligned} \tag{6}$$

Again each term in this sum is non-negative and must therefore be zero, but this contradicts $j_1, \ldots, j_k \notin I$. Hence edges with $i \in I$ and $j_1, \ldots, j_k \notin I$ do not exist, but this contradicts the assumption of irreducibility. Since $I \neq \emptyset$, we must have $I = \mathcal{V}$ or $x = y$.

Next consider directed hypergraphs with hyperedges $E = (S,T)$, $S,T \subset \mathcal{V}$ as defined before. Then, define $\mathcal{R}_{p,q}(x,y)$ for $x,y \in \mathbb{R}^N$ and $p,q \geqslant 1$,

$$\mathcal{R}_{p,q}(x,y) = \sum_{(S,T)\in\mathcal{E}} w(S,T) \prod_{i\in S} \left(\frac{|x_i|}{\|x\|_p}\right)^{\frac{1}{2|S|}} \prod_{j\in T} \left(\frac{|y_j|}{\|y\|_q}\right)^{\frac{1}{2|T|}}. \tag{7}$$

By identical arguments as for undirected hypergraphs, it can be shown that for a strongly connected directed hypergraph, there exists a unique pair $x \in \mathbb{S}_p^N$ and $y \in \mathbb{S}_q^N$ such that $\mathcal{R}_{p,q}(x,y) \geqslant \mathcal{R}_{p,q}(x',y')$ for all $x',y' \in \mathbb{R}^N$. These maximizers are strictly positive and satisfy the Euler-Lagrange equations

$$\lambda_{p,q} x_i^p = \sum_{\{(S,T)\in\mathcal{E}: i\in S\}} \frac{w(S,T)}{2|S|} \left(\prod_{i'\in S} x_{i'}\right)^{\frac{1}{2|S|}} \left(\prod_{j\in T} y_j\right)^{\frac{1}{2|T|}}, \tag{8}$$

$$\lambda_{p,q} y_j^q = \sum_{\{(S,T)\in\mathcal{E}: j\in T\}} \frac{w(S,T)}{2|T|} \left(\prod_{i\in S} x_i\right)^{\frac{1}{2|S|}} \left(\prod_{j'\in T} y_{j'}\right)^{\frac{1}{2|T|}}, \tag{9}$$

subject to the constraints $\|x\|_p = \|y\|_q = 1$ and with $\lambda_{p,q} = \mathcal{R}_{p,q}(x,y)$. Details are given in Appendix B.

V. SPECTRAL CLUSTERING AND BICLUSTERING IN HYPERGRAPHS

Having a generalization of the Perron-Frobenius theorem, it is straightforward to also generalize the spectral clustering method. Define, for $X \subset \mathcal{V}$,

$$\mathcal{S}_p(X) = \frac{\sum_{E\subset X} w(E)}{|X|^{\frac{1}{p}}} = \mathcal{R}_p(u_X) \leqslant \mathcal{R}_p(x), \tag{10}$$

with x the dominant eigenvector and $u_X \in \mathbb{S}_p^N$ now defined by $u_{X,i} = |X|^{-1/p}$ for $i \in X$, and 0 otherwise. The parameter p balances cluster size versus edge density. For $p = 1$, $\mathcal{S}_p$ is the ratio of edges to nodes in X. Taking $p > 1$ diminishes the influence of the denominator and progressively favors a high number of edges rather than a high number of edges per node in high-scoring clusters (further details are given in Sec. VII). The spectral clustering algorithm becomes as follows:

(1) Calculate the maximizer x of $\mathcal{R}_p$.

(2) Find the cluster $X_{\max}$ which solves the restricted variational problem

$$X_{\max} = \underset{c>0}{\operatorname{argmax}}\, \mathcal{S}_p(X_c),$$

with $X_c = \{i \in \mathcal{V} : x_i > c\}$.

(3) Store $X_{\max}$, remove all hyperedges between nodes in $X_{\max}$ from the edge set $\mathcal{E}$, and repeat the procedure until no more hyperedges remain.

The maximizer can be calculated using a generalization of the power method for matrices [19] or tensors [22]: starting with an initial vector $x^{(0)}$ and defining $\lambda_p^{(0)} = \|x^{(0)}\|_p = 1$, we compute $x^{(n+1)}$ from $x^{(n)}$ using the Euler-Lagrange equations (4) in the following steps:

$$x_i^{(n+1)} \leftarrow \left[\sum_{\{E\in\mathcal{E}:i\in E\}} \frac{w(E)}{|E|} \left(\prod_{j\in E} x_j^{(n)} \right)^{\frac{1}{|E|}} \right]^{\frac{1}{p}}, \tag{11}$$

$$\lambda_p^{(n+1)} = \|x^{(n+1)}\|_p, \tag{12}$$

$$x_i^{(n+1)} \leftarrow \frac{x_i^{(n+1)}}{\lambda_p^{(n+1)}}, \tag{13}$$

iterated until the components of $x^{(n)}$ become stationary or, equivalently, $\lambda_p^{(n)}$ has converged to the dominant eigenvalue, i.e.,

$$\left| 1 - \frac{\lambda_p^{(n+1)}}{\lambda_p^{(n)}} \right| < \epsilon, \tag{14}$$

where ϵ is a predefined numerical tolerance threshold. Due to the uniqueness of x, the choice of starting vector is not important. By taking a non-negative one, such as the uniform vector $x^{(0)} = [1,1,\dots,1]^T/N^{1/p}$, we ensure that the powers of $1/|E|$ occurring in the Euler-Lagrange equations are always defined unambiguously. Many of the hypergraphs occurring in real-world applications are not connected. In such cases, it is important to ensure that $x^{(0)}$ has support only on a single connected component to obtain the unique maximizer for that component.

Although we typically view a cluster as a subset of vertices, it is actually a subset of hyperedges (all hyperedges $E \subset X_{\max}$) and thus can be considered as a subhypergraph as well. Higher-scoring clusters can thus be obtained by recursively applying the previous procedure to each of the clusters itself until no more subdivision that improves the score is found.

For directed hypergraphs, we have a biclustering method. Define, for $X,Y \subset \mathcal{V}$ and $p,q \geqslant 1$,

$$\mathcal{S}_{p,q}(X,Y) = \frac{\sum_{S\subset X,T\subset Y} w(S,T)}{|X|^{\frac{1}{2p}}|Y|^{\frac{1}{2q}}}.$$

Approximate maximizers $X_{\max}$ and $Y_{\max}$ are found by solving the restricted variational principle,

$$(X_{\max},Y_{\max}) = \underset{(c_1,c_2)}{\operatorname{argmax}}\, \mathcal{S}_{p,q}\big(X_{c_1},Y_{c_2}\big),$$

with $X_{c_1} = \{i \in \mathcal{V} : x_i > c_1\}$ and $Y_{c_2} = \{i \in \mathcal{V} : y_i > c_2\}$, where x and y are the unique solutions of the Euler-Lagrange equations (8) and (9), which can again be calculated using a power algorithm.

VI. RELATION TO PREVIOUS WORK

The matrix algorithm for clustering in a simple graph has its roots in a method for image pattern recognition [23], and the use of the singular value decomposition to detect densely linked sets in directed networks goes back to the work of Kleinberg [24]. The novelty here lies in the definition of a discrete cluster through solving the restricted variational problem, instead of using an *ad hoc* cutoff on the eigenvector entries. For k-uniform hypergraphs, we can define rescaled variables $y_i = x_i^{1/k}$ such that maximizing $\mathcal{R}_p(x)$ becomes equivalent to maximizing

$$\mathcal{R}'_{p'}(y) = \frac{\sum_{i_1,\dots,i_k} T_{i_1,\dots,i_k} y_{i_1},\dots,y_{i_k}}{\|y\|_{p'}^k},$$

with $p' = kp$. In this case, the Theorem presented in Sec. IV reduces to a multilinear extension of the Perron-Frobenius theorem to non-negative irreducible tensors of arbitrary dimension, which has been the subject of several recent papers [25–27] (which all depend on the strong irreducibility condition). The Proof given in Sec. IV is considerably simpler, holds for general connected hypergraphs, and follows more closely the proof of the matrix theorem [18]. In the unscaled variational problem for $\mathcal{R}'_{p'}$, the maximizer is unique for $p' \geqslant k$ and thus it is unsuited for generalizing to arbitrary hypergraphs where the uniqueness condition would become $p' \geqslant k_{\max}$, which is the maximum edge size in the hypergraph. This explains why we introduced the geometric average over the values x_j in Eq. (3).

For $k = 3$, we have previously used a similar approach to find clusters of three-node network motifs in integrated interaction networks [28,29]. In this case, an adjacency tensor T_{rst} is defined to be 1 if an instance of a three-node query motif or graph pattern exists between vertices (r,s,t), and 0 otherwise. More generally, we can define for any k-node query pattern a k-uniform hypergraph consisting of all instances of the query pattern in a given graph $\mathcal{G}$. Our algorithm will identify clusters of vertices in $\mathcal{G}$ with a high number of pattern instances between them, which often have a functional meaning in biological networks [28,30].

Another example for $k = 3$ concerns the analysis and clustering of multiply linked data [31,32] or multislice networks [33]. Here we are given a set of M directed or undirected graphs and define a hypergraph adjacency tensor as $T_{ijm} = A_{ij}^{(m)}$, where $A^{(m)}$ denotes the adjacency matrix of the mth graph. Clustering in this case identifies vertex sets which are densely connected in multiple, but not necessarily all, graphs.

VII. ALGORITHM VALIDATION

A. Random geometric graphs

The dominant eigenvector of a graph's adjacency matrix is often considered as a centrality measure ("eigenvector centrality" [1]) and is, in essence, equal to a simplified PageRank [34] for ranking global vertex importance. It may thus come as a surprise to see it playing a role in identifying localized clusters (however, see the references in the previous section). In order to demonstrate the validity of our approach and illustrate the statements in Sec. V, we applied it to randomly generated geometric graphs of various sizes (see Appendix C 1 for details).

For visualization purposes, we generated as a toy example a random geometric graph with 100 vertices and radius $r^2 = 0.02$ [Fig. 1(a)]. The graph is evidently modular and the six highest-scoring edge clusters identified by our algorithm (with $p = 1$) are indicated in color. The profiles of the corresponding dominant eigenvectors are clearly localized on a subset of nodes [Fig. 1(b)], illustrating that in a modular network, the dominant eigenvector indeed indicates the location of a single cluster. Furthermore, comparing the edge-to-node ratio for each of the discovered edge clusters with the theoretical upper bound in Eq. (1) shows that the solution of the restricted variational problem [Eq. (2)] must be close to the true maximum [Fig. 1(c)].

For a more systematic analysis, we performed triangle-based clustering on sequences of geometric graphs with constant expected edge density and varying size. Triangle-based clustering searches for overlapping sets of triangles in an ordinary graph and corresponds to the simplest form of k-clique clustering [35]. Here we considered each instance of a triangle in the input graph as a hyperedge in a three-uniform hypergraph to which we applied our spectral clustering algorithm. The parameter p can be used to identify clusters at different levels of resolution. Independent of network size, there is a low-p phase where the fraction of nodes in a cluster is small compared to total network size, and a high-p phase where a cluster consists of a macroscopic network portion [Fig. 1(d)]. Interestingly, at $p = 1$, cluster size does not depend on network size [Fig. 1(d), inset]. Hence clustering based on (hyper)edge-to-node ratio scores [Eq. (10)] does not suffer from a resolution limit problem where cluster size grows with network size irrespective of the presence of "natural" clusters at smaller scales [36,37]. As in the previous example, the cluster scores are always close to their theoretical upper bounds, demonstrating that the solution of the restricted variational problem is close to the true optimum in all cases (see Fig. S1 of Supplemental Material [38]).

B. Edge-to-node scaling parameter

The transition in Fig. 1(d) as a function of the edge-to-node scaling parameter p is a general feature, independent of the actual hypergraphs used, and can be easily understood as follows. Assume we have a hypergraph $\mathcal{H} = (\mathcal{V},\mathcal{E})$ with $N = |\mathcal{V}|$ nodes and $M = |\mathcal{E}|$ hyperedges. Then the relative score of any set $X \subset \mathcal{V}$ with $n = |X|$ nodes and m hyperedges compared to the score of the total hypergraph is

$$\frac{\mathcal{S}_p(X)}{\mathcal{S}_p(\mathcal{V})} = \frac{m}{M}\left(\frac{N}{n}\right)^{\frac{1}{p}} = \frac{\alpha_2}{\alpha_1^{1/p}} \equiv s_p(\alpha_1,\alpha_2),$$

with α_1 and α_2 the fractions of nodes and edges in X. The phase diagram of s_p as a function of these two variables is independent of the actual hypergraph under consideration (Fig. 2). Naturally, not all combinations of α_1 and α_2 are admissible. In general, there exists a boundary $\alpha_2 \leqslant f(\alpha_1)$ with $f(\alpha_1) \approx \alpha_1$ for $\alpha_1 \approx 1$. In sparse hypergraphs, we typically have $M \sim N^{1+\delta}$ with δ small, where often $\delta = 0$. Locally, however, the edge density can be much higher. For instance, in ordinary edge clustering, $m \sim n^2$, and in triangle-based clustering, $m \sim n^3$, for n not too large. Hence, as α_1 decreases from 1, the boundary function $f(\alpha_1)$ will deviate more and more from the diagonal $\alpha_2 = \alpha_1$. In Fig. 2, we have sketched a typical shape of a boundary function (thick line). At $p = 1$ (Fig. 2, top left), the contour lines of s_p are straight lines and s_p will clearly be maximal at small values of (α_1,α_2). As p increases, the contour lines become increasingly more concave, pushing the value where s_p attains its maximum towards $\alpha_1 = 1$. For the idealized boundary function in Fig. 2, the transition is in fact discontinuous and jumps from being at $\alpha_1 = 0.1$ (origin of the axes) to $\alpha_1 = 1$ around $p = 1.95$ (bottom left).

Since the transition is, in general, sharp as a function of p and can even be discontinuous, we will in practice only use the default edge-to-node ratio score with $p = 1$ to identify dense hypergraph clusters, or use a large value of p (typically $p \gtrsim 10$) to identify connected hypergraph components.

C. Algorithm efficiency

For an undirected hypergraph with N nodes, M hyperedges, and maximum edge size $k_{\max}$, the update steps in the power algorithm are at most of the order $k_{\max}M$ [Eq. (11)] and N [Eqs. (12) and (13)]. The number of steps needed to reach convergence depends on the convergence parameter ϵ [Eq. (14)] and therefore possibly also on the hypergraph size. In practice, a maximal number of iterations $I_{\max}$ is defined and convergence manually inspected when $I_{\max}$ is exceeded. Determining the optimal threshold value is, at most, of the order of N (number of possible threshold values) times M (calculation of the edge-to-node ratio score). Taken together, runtime is bounded by

$$t_{\text{run}} \leqslant I_{\max}[O(k_{\max}M) + O(N)] + O(MN).$$

For directed hypergraphs, determining the optimal threshold pair over all possible combinations of entries of the dominant singular vector pair (x,y) is of the order of N^2M,

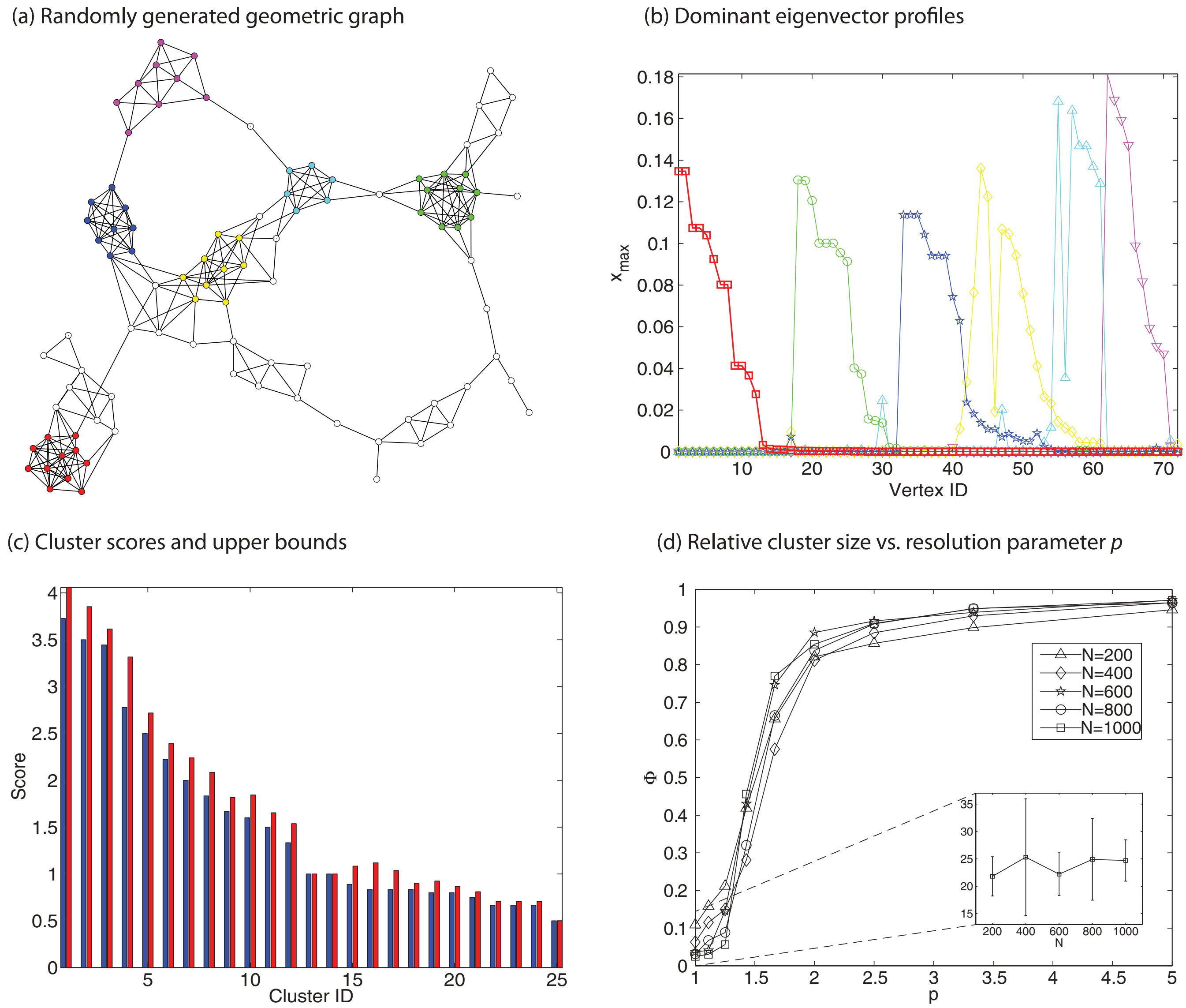


FIG. 1. (Color online) (a) Example of a randomly generated geometric graph with 100 vertices and radius $r^2 = 0.02$, showing the largest connected component with the six highest-scoring edge clusters indicated by filled nodes. (b) Dominant eigenvector profiles for the six highest-scoring edge clusters. (c) Edge-to-node ratio scores (left blue bars) and theoretical upper bound (right red bars) for all 25 edge clusters. (d) Cluster size as the fraction Φ of total number of network nodes for the highest-scoring triangle-based cluster in random geometric graphs with $N = 200$, 400, 600, 800, and 1000 nodes and constant edge density ($\rho = 4$) as a function of p. Each data point is an average over 10 random networks. The inset shows the absolute mean cluster size and standard deviation over 10 random networks as a function of N for $p = 1$.

which is often prohibitive. In such instances, taking

$$X_{\max} = \underset{c}{\operatorname{argmax}}\, \mathcal{R}_{p,q}(u_{X_c}, y),$$

$$Y_{\max} = \underset{c}{\operatorname{argmax}}\, \mathcal{R}_{p,q}(x, u_{Y_c}),$$

where we used the same notation as in Sec. V, results in an approximation which is again $O(MN)$.

VIII. APPLICATIONS

A. Local and global alignment of complex networks

The core idea for applying hypergraph clustering to the analysis of edge-colored graphs is to translate the relation between multiple interaction types (edge colors) into higher-order hypergraph edges. We illustrate this idea by showing that local and global alignment of complex networks with a bipartite many-to-many mapping between their vertex sets can be naturally viewed as a hypergraph clustering problem.

Network alignment is the problem of finding topologically similar regions between two or more networks. In local network alignment, small subgraphs in each network are aligned independently of the alignment of other subgraphs, whereas global network alignment aims to find a maximal alignment for each connected component in the input graphs. Network alignment methods for comparing molecular interaction networks between different species come in two main flavors. Topological network alignment finds conserved regions between networks taking only the topology of each

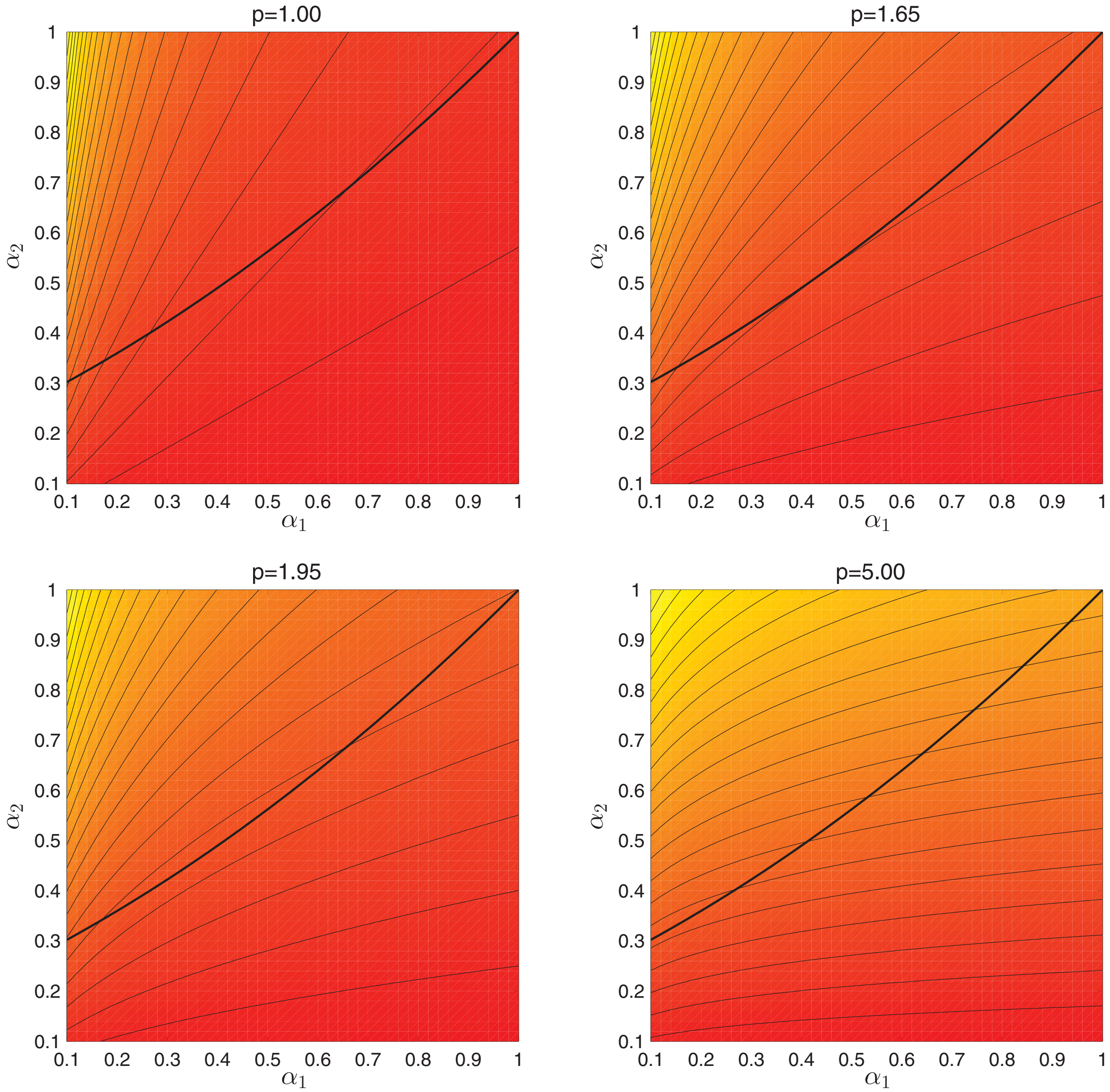
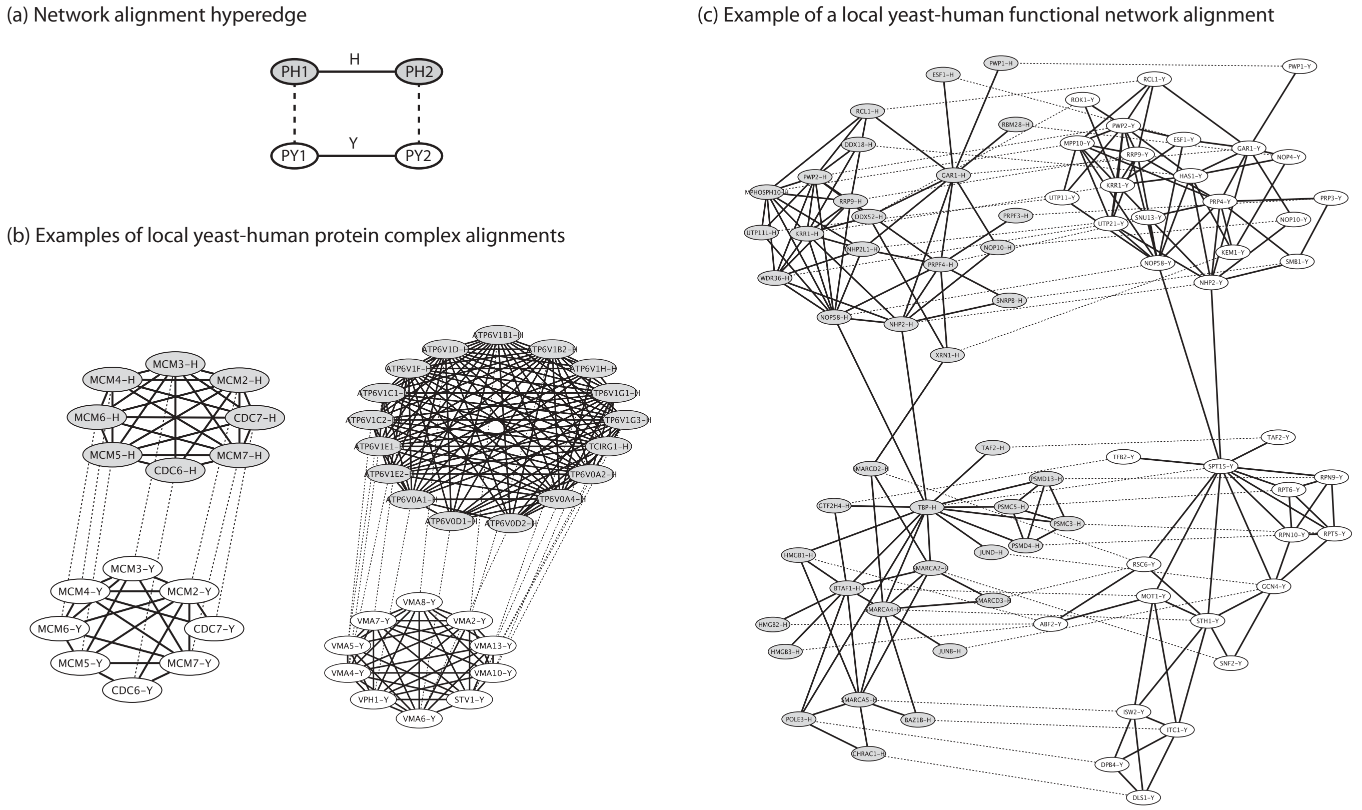


FIG. 2. (Color online) Phase diagrams of $s_p(\alpha_1,\alpha_2)$ for $p = 1,1.65,1.95$, and 5 (left to right, top to bottom). More yellow (lighter gray) indicates higher values of s_p; the thin lines are contours of constant s_p, while the thick line indicates a possible boundary of admissible states. Colors (gray scale levels) are relative to the minimum and maximum in each panel and not comparable between panels.

network into account [39]. The second class of methods takes into account that networks in different species have evolved from a common ancestor through gene duplication and divergence mechanisms and hence there exists a meaningful mapping between the nodes in each network [12]. Methods have been developed which assume a one-to-one mapping [40], but in general a many-to-many map should be considered [41].

More formally, consider two ordinary graphs $\mathcal{G}_1$ and $\mathcal{G}_2$, whose vertices are connected by a bipartite graph $\mathcal{M}$. The directed alignment hypergraph $\mathcal{H}$ between $\mathcal{G}_1$ and $\mathcal{G}_2$ is defined as the four-uniform hypergraph containing the edges $(\{i,j\},\{k,l\})$, if and only if $\{i,j\} \in \mathcal{G}_1$, $\{k,l\} \in \mathcal{G}_2$, and $\{i,k\},\{j,l\} \in \mathcal{M}$ [Fig. 3(a)]. Such alignment hyperedges are also called *interologs*. Interolog mapping is routinely used to transfer annotation information from one organism to another [42] and interolog analysis is at the heart of previous network alignment methods [41,43]. Here we propose to address the network alignment problem by identifying hyperedge clusters in the alignment or interolog hypergraph. Indeed, in a local alignment, we search for small regions in each graph which map nearly perfectly onto each other, i.e., have a high density of interologs between them. This corresponds to hypergraph clusters which maximize $\mathcal{S}_p$ for values of p close to one. In a global alignment, we search for maximally matching regions in each graph, i.e., connected components in the interolog



(a) Network alignment hyperedge

(b) Examples of local yeast-human protein complex alignments

(c) Example of a local yeast-human functional network alignment

FIG. 3. (a) A (directed) hyperedge in the yeast-human protein interaction network alignment hypergraph is an *interolog*: a pair of interacting yeast (Y) proteins and a pair of interacting human (H) proteins connected by orthology relations (dashed lines). (b) Examples of aligned protein complexes (cluster no. 19 left, no. 1 right). (c) Example of a functional network alignment (cluster no. 48). In all panels, yeast proteins are white and human proteins are gray; protein interactions are solid lines and orthology relations are dashed lines.

hypergraph. These correspond to hypergraph clusters which maximize $\mathcal{S}_p$ for large values of p.

We used our spectral clustering algorithm to locally and globally align protein-protein interaction networks between yeast and human, using orthology groups for mapping conserved proteins between both organisms (see Appendix C 2 for details). Protein-protein interaction networks represent binary, undirected associations between proteins and they are, at present, the most extensively characterized molecular interaction networks in biology [11,44]. Typical examples of high-scoring local alignment clusters are conserved protein complexes (see Supplemental Material [38]). Figure 3(b) shows two examples: first is a set of proteins that maps one-to-one between yeast and human from the minichromosome maintenance (MCM) complex (cluster no. 19), which plays an important role in DNA replication and is indeed conserved among all eukaryotes [45]; and second (cluster no. 1) is a set of components of the V-type ATPase (a proton pump), which has expanded in human compared to yeast by gene duplications [46]. Other local alignment clusters reflect more general functional networks than protein complexes (see Table S3 of Supplemental Material [38]). Figure 3(c) shows cluster no. 48, which is an example of a conserved network involved in nucleic acid metabolism centered around the general transcription factor TBP (SPT15 in yeast), i.e., the TATA-binding protein. The largest connected component in the network alignment hypergraph maps 651 yeast proteins to 766 human proteins and contains 90% of all interologs (see Table S4 of Supplemental Material [38]), showing that there exists a high degree of network conservation at a global scale, which is consistent with previous findings using topological network alignment [39].

B. Tripartite community detection in online folksonomies

Folksonomies, i.e., online communities where users collaboratively create and annotate data, are examples of social systems that cannot be adequately modeled by ordinary graphs. For instance, tagged social networks such as Flickr [47] or CiteULike [48] have a tripartite structure that is best modeled by a three-uniform hypergraph [8,9]. Using CiteULike as a concrete example, each hyperedge consists of a user who has annotated an academic article with a certain keyword or tag [48] [Fig. 4(a)]. Traditionally, the community structure of such tripartite networks has been analyzed by considering one-mode ordinary graph projections of the hypergraph, e.g., by connecting two users if they have annotated the same articles or connecting two tags if they have been applied to the same articles [9]. In contrast, hypergraph-based clustering preserves the tripartite structure of folksonomy data and reveals additional levels of community structure. We applied our spectral clustering algorithm to a subset of the CiteULike data set containing more than 400 000 (user, article, tag) entries and identified nearly 14 000 hyperedge clusters (see

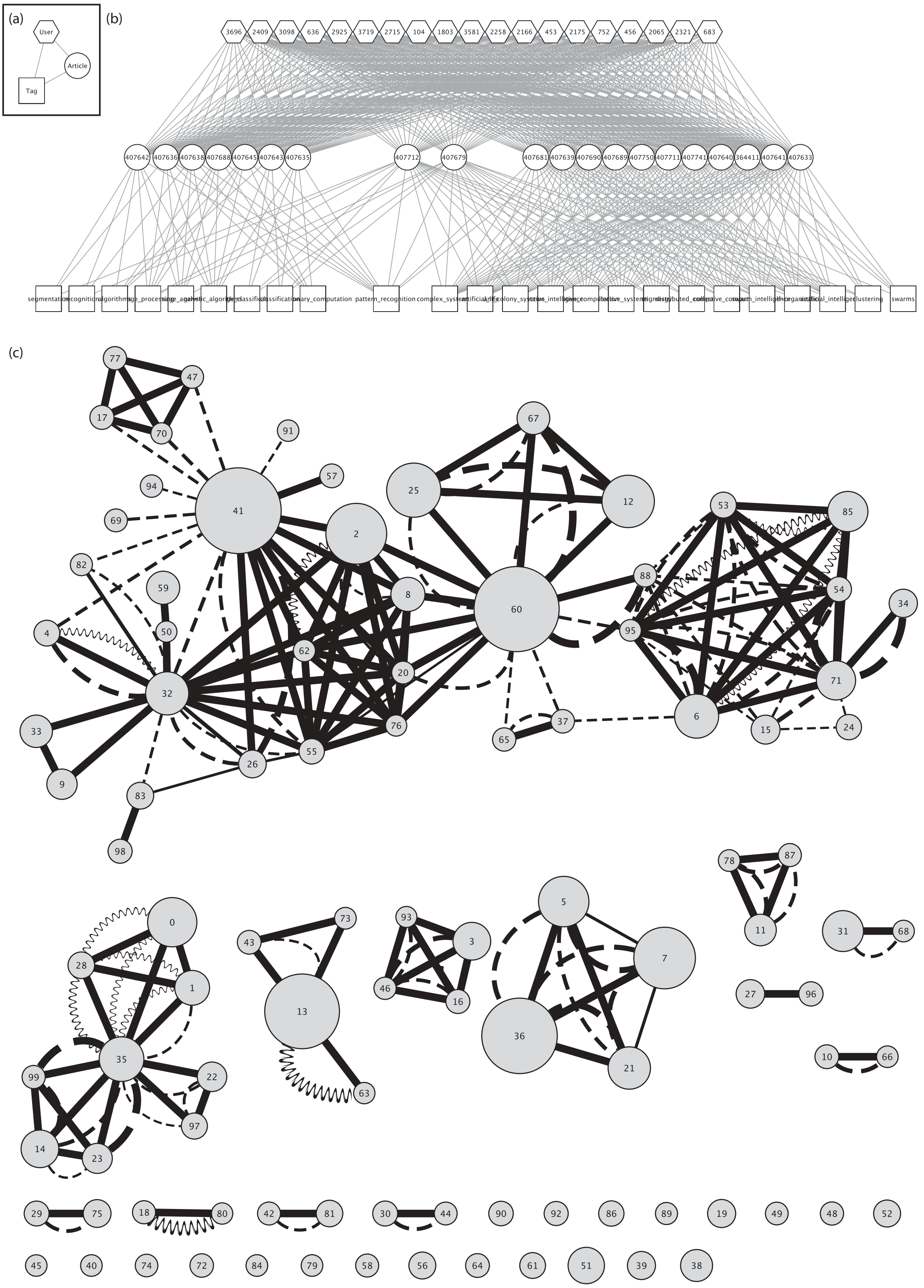


FIG. 4. (a) CiteULike hyperedge, which represents one instance of a user (hexagonal node) who has annotated an article (circular node) with a certain tag (rectangular node). (b) Example of two tripartite communities where the same set of users (top) has annotated two sets of articles (middle) with two sets of tags (bottom). Only the two central articles and one central tag (“pattern recognition”) overlap between the two clusters. User-tag edges have been omitted for clarity. (c) Coarse-grained view of the CiteULike hypergraph using the 100 highest-scoring hyperedge clusters. Each node represents a cluster (with node size proportional to the number of hyperedges in the cluster) and edges represent significant overlap between clusters (overlap score > 0.5; edge size proportional to overlap score). Solid edges: user overlap; dashed edges: tag overlap; wavy edges: article overlap.

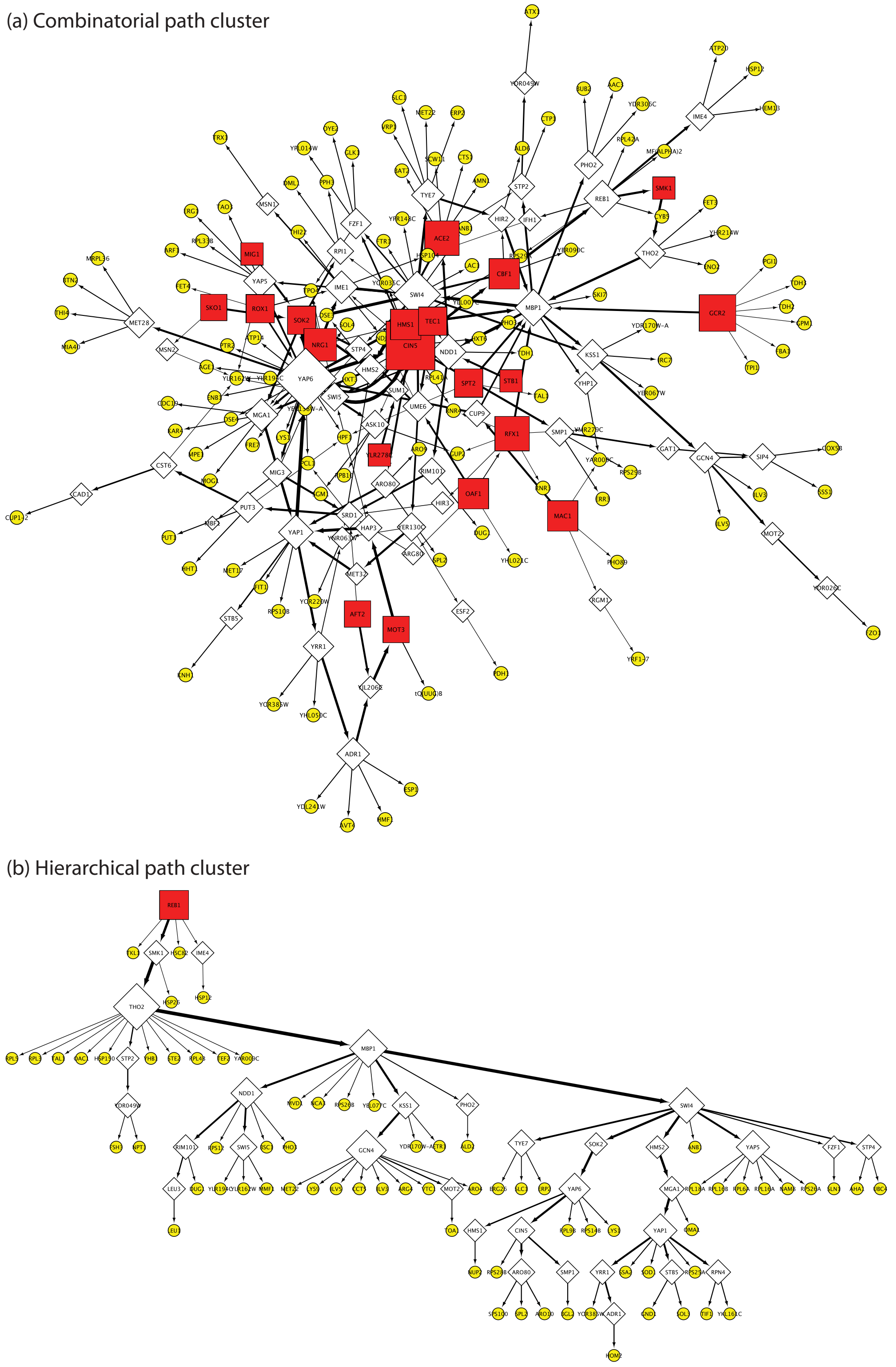


FIG. 5. (Color online) Examples of high-scoring combinatorial [(a), cluster no. 6] and hierarchical [(b), cluster no. 1] path clusters in the yeast transcriptional regulatory network. Red (dark gray) rectangular nodes: knocked-out transcription factors (TFs); yellow (light gray) circular nodes: genes differentially expressed upon knock out of the TFs; white diamond-shaped nodes: all other TFs. Node size is proportional to out-degree and edge width to edge "betweenness" (defined for the purposes of this figure as the number of shortest paths between all pairs of cluster nodes passing through a given edge).

Appendix C 3 for details). The additional level of detail present in hyperedge clusters is illustrated by looking at the user, article, or tag overlap between clusters. Figure 4(b) shows an example of two hyperedge clusters formed by the same set of users who have annotated different sets of articles by different sets of tags. Only one tag, i.e., "pattern recognition," is common between both clusters. The remaining tags show that the articles in the first cluster are about collective computing and swarm intelligence, whereas those in the second cluster deal with image analysis (see Table S1 of Supplemental Material [38]), which are indeed two distinct subjects within the broad field of pattern recognition.

In general, we expect such subdivisions of one-mode projected communities to occur at the level of users (i.e., the same set of users annotating different sets of articles using different sets of tags), but much less at the level of articles or tags (i.e., we do not expect different sets of users to annotate the same set of articles using different sets of tags, or to use the same set of tags for different sets of articles). Indeed, the 100 highest-scoring clusters (which together contain about 20% of all hyperedges) overlap predominantly at the user level, to a much lesser extent at the tag level, and hardly overlap at the article level, while about 21 of these clusters do not have any significant overlap (overlap $> 50\%$; see Appendix C 3 for details) with any other cluster [Fig. 4(c)]. Significant article overlap occurs in only two instances. In both cases, it concerns a subset of users who have annotated a subset of articles from a larger cluster with an additional set of tags. Tag overlap occurs more frequently than article overlap, but with lower overlap percentages than user overlaps. Overlapping tags are typically general tags which can be applied to a broad spectrum of articles. For instance, the ten tags occurring most frequently in the top 100 clusters are bibtex-import, learning, social, evolution, review, support, govt, non-us, collaboration, and design. Thus we conclude that hyperedge clusters capture topic-specific tripartite (user, article, tag) communities which reveal more structure of the underlying data than user, article, or tag communities based on a single data dimension only.

C. Path clustering in regulatory networks

Unlike protein-protein interaction networks, which are undirected, regulatory networks, which control the cellular response to external or internal perturbations, are directed and represent the flow of information within a cell [10]. In transcriptional regulatory networks, the response to perturbations can be measured experimentally by genetically knocking out a transcription factor (TF) and measuring the resulting changes in gene expression levels on a genome-wide scale [49]. In yeast, direct physical binding interactions between a TF and its target genes [50] as well as perturbational response data for the same TF [49] are available for a comprehensive set of almost 200 TFs (see Appendix C 4 for details). On average, only 3% of the genes which respond to a knock-out perturbation of a TF are also direct physical targets of that TF, and various approaches have been proposed to understand the mechanisms of indirect regulation and propagation of network perturbations in this context [51–54]. It is thought that perturbational responses are organized in a modular way, in the sense that groups of genes will be affected by the knock out of a TF through the same intermediate regulatory pathways. However, due to the variable length of these pathways, previous approaches for clustering in directed networks (e.g., [55–57]), which identify densely interacting node sets, are not directly applicable to this problem.

Here we address the problem of identifying sets of nodes which respond to the knock out of a TF through similar regulatory paths by defining a nonuniform hypergraph where each hyperedge corresponds to a shortest path between two nodes in the original regulatory network. Hypergraph-based clustering will then find sets of nodes with a high number of shortest paths running through them and such clusters form potential "signal-propagation" modules, which is consistent with the notion that high information flow in a network is associated to high values of a node's "betweenness" centrality (defined as the number of shortest paths between all pairs of nodes passing through a given node). To test this hypothesis, we calculated all directed shortest paths in the regulatory network of yeast between a TF and the genes differentially expressed upon knock out of that TF. The resulting hypergraph contained 1332 hyperedges between 788 nodes, and spectral clustering identified 25 nonsingleton and 14 singleton clusters (see Appendix C 4 for details). Topologically, there appear to exist two distinct types of path clusters. Combinatorial path clusters contain genes responding to the knock out of multiple TFs and form a network of densely overlapping paths. Figure 5(a) shows a combinatorial cluster of 199 shortest paths from 20 TFs to 186 genes involved in glycolysis and gluconeogenesis. Hierarchical path clusters have a layered structure, where the perturbational signal of usually not more than one TF flows to its targets via a limited number of intermediate TFs, in a strictly hierarchical manner [Fig. 5(b)]. The functional relevance of regulatory path clusters is demonstrated by the fact that they contain a significant fraction of the genes affected by the deletion of the cluster's TF and that they strongly overlap with specific functional categories (see Tables S5 and S6 of Supplemental Material [38]). For simplicity, we considered here only the shortest paths in the transcriptional regulatory network, but clearly the approach can be extended to paths composed of multiple interaction types.

IX. CONCLUSIONS

Over the past decade, graph theory has become crucial to represent and reason about complex network data. In particular, clustering, i.e., the detection of densely interconnected groups of vertices with few connections to the rest of the network, has become a standard coarse-graining procedure to understand the structure and function of complex networks. With more and more data becoming available to highlight different aspects of the same complex systems, a need has arisen to analyze networks with multiple types of interactions simultaneously. In this paper, we have proposed to use hypergraphs to characterize higher-order relations between simple graphs and we have introduced efficient algorithms for clustering and biclustering in such hypergraphs.

Our main result is a spectral clustering algorithm for hypergraphs, based on a generalization of the Perron-Frobenius theorem for directed and undirected hypergraphs. More precisely, we have shown that like in ordinary graphs, there exists a unique, positive vector, called the dominant eigenvector, over the set of vertices of a hypergraph, which maximizes a natural generalization of the Rayleigh-Ritz ratio for matrices. The importance of this result lies in the fact that the ratio of the number of edges to the number of nodes in any subset of vertices can be expressed as the same Rayleigh-Ritz ratio, in graphs and hypergraphs alike. Densely interconnected clusters can therefore be found very efficiently by first computing the dominant eigenvector and then converting it to a discrete set of vertices. Uniqueness of the dominant eigenvector guarantees unambiguity of the solution and rapid convergence of the numerical procedure, whereas positivity implies that the discretization can be achieved by setting an optimal threshold on its entries.

Our work has been motivated by concrete problems of data integration in social and biological networks. We have given three practical examples for using hypergraph-based clustering in these contexts, namely, the alignment of protein-protein interaction networks between multiple species using interolog clustering, the detection of tripartite communities in folksonomies, and the identification of overlapping regulatory pathways in perturbational expression data using shortest path clustering. Undoubtedly, many more applications for hypergraph-based clustering exist in the analysis of other biological, social, computer, communication, or neural networks. From a theoretical point of view, we have considered the edge-to-node ratio as a simple quality score for clusters in graphs and hypergraphs. Although this score has many attractive properties, such as its direct relation with the dominant eigenvector and the absence of any resolution limit problems, it will still be of interest to generalize clustering algorithms based on other quality scores from graphs to hypergraphs as well. Popular methods like those based on minimal cutsets or modularity maximization also rely on spectral properties of, respectively, the graph Laplacian and modularity matrix. Although certain mathematical aspects, such as eigenvalue multiplicity and its implications on algorithm convergence and cluster discretization, are more complicated in these cases, we believe our work lays the theoretical foundations for future studies in this direction.

ACKNOWLEDGMENTS

T.M. thanks the Department of Mathematics of the University of California, Davis for warm hospitality during visits when part of this work was performed. The work of B.N. was supported in part by the National Science Foundation under Grant No. DMS-1009502. We thank the referees for their detailed and constructive comments which led to a much improved version of our paper.

APPENDIX A: STRONG CONNECTIVITY OF DIRECTED HYPERGRAPHS

Consider first an undirected hypergraph $\mathcal{H} = (\mathcal{V},\mathcal{E})$ on N vertices. Although connectedness of $\mathcal{H}$ does not imply irreducibility, we do have the property that if there exists a proper subset $I \subset \mathcal{V}$ such that for all $i_1,\ldots,i_k \in I$ and $j_1,\ldots,j_m \notin I$, $w(\{i_1,\ldots,i_k,j_1,\ldots,j_m\}) = 0$, then $\mathcal{H}$ is not connected (since there can then be no path that starts in I and escapes from I). Hence, if $\mathcal{H}$ is connected, no such set I exists.

For a directed hypergraph $\mathcal{H} = (\mathcal{V},\mathcal{E})$, we can define an underlying undirected hypergraph $\tilde{\mathcal{H}} = (\mathcal{V},\tilde{\mathcal{E}})$ by considering all possible partitions of a subset $E \subset \mathcal{V}$ into source and target sets, i.e., $\tilde{w}(E) = \sum_{\{(S,T):S\cup T=E\}} w(S,T)$. This procedure generalizes the definition of a symmetric adjacency matrix $B = A + A^T$ from the asymmetric adjacency matrix A of a directed graph. Clearly, to call $\mathcal{H}$ connected, we shall ask that $\tilde{\mathcal{H}}$ is connected as defined in Sec. II.

Now consider two subsets $I,J \subset \mathcal{V}$ such that $I \cup J$ is neither empty nor equal to $\mathcal{V}$. Since $\tilde{\mathcal{H}}$ is connected, there exist vertices $i_1,\ldots,i_k \in I$, $j_1,\ldots,j_\ell \in J$, and $h_1,\ldots,h_m \notin I \cup J$ such that

$$\tilde{w}(\{i_1,\ldots,i_k,j_1,\ldots,j_\ell,h_1,\ldots,h_m\}) > 0.$$

This implies that there exists at least one partition of these nodes into a source and target set with nonzero directed weight. We ask slightly more, namely, that there is a partition of the form

$$w(\{i_1,\ldots,i_k,h_1,\ldots,h_n\},\{j_1,\ldots,j_\ell,h_{n+1},\ldots,h_m\}) > 0,$$

i.e., the source as well as the target set should contain at least one element not in I or J. Note that the requirement that all i's go into the source set and all j's go into the target set is purely notational convenience, since I or J are allowed to be empty, as long as their union is not. If the above condition is fulfilled for all pairs of sets (I,J), we say that the directed hypergraph $\mathcal{H}$ is *strongly connected.*

APPENDIX B: GENERAL PROOF OF THE PERRON-FROBENIUS THEOREM FOR CONNECTED HYPERGRAPHS

Consider a non-negative maximizer x of $\mathcal{R}_p(x)$ and without loss of generality assume $\|x\|_p = 1$. Let again $I = \{i \in \mathcal{V} : x_i = 0\}$ and assume $I \neq \emptyset$. Let k be the smallest integer for which there exists at least one set $i_1,\ldots,i_k \in I$ and at least one set $j_1,\ldots,j_m \notin I$ such that $w(\{i_1,\ldots,i_k,j_1,\ldots,j_m\}) > 0$. Such k must exist, since $\mathcal{H}$ is connected (see Appendix A). For $\epsilon > 0$, define

$$\tilde{x}_i = \begin{cases} x_i & i \notin I \\ \epsilon & i \in I. \end{cases}$$

We will show that for ϵ small enough, $\mathcal{R}_p(\tilde{x}) > \mathcal{R}_p(x)$, which contradicts the assumption that there can exist a maximizer with zero elements. We have

$$\|\tilde{x}\|_p^p = \|x\|_p^p + |I|\epsilon^p = 1 + |I|\epsilon^p,$$

or, to leading order in ϵ,

$$\frac{1}{\|\tilde{x}\|_p} = 1 - \frac{|I|}{p}\epsilon^p + o(\epsilon^p). \tag{B1}$$

For the denominator of $\mathcal{R}_p(\tilde{x})$, we have

$$\begin{aligned} \sum_{E\in\mathcal{E}} w(E)\left(\prod_{i\in E}\tilde{x}_i\right)^{\frac{1}{|E|}} &= \sum_{\{E\in\mathcal{E}:E\cap I=\emptyset\}} w(E)\left(\prod_{i\in E}x_i\right)^{\frac{1}{|E|}} \\ &\quad + \sum_{\{E\in\mathcal{E}:E\cap I\neq\emptyset\}} w(E)\left(\prod_{i\in E}\tilde{x}_i\right)^{\frac{1}{|E|}} \\ &= \mathcal{R}_p(x) + \sum_{\{E\in\mathcal{E}:E\cap I\neq\emptyset\}} w(E)\left(\prod_{i\in E}\tilde{x}_i\right)^{\frac{1}{|E|}}. \end{aligned} \tag{B2}$$

From the preceding discussion, it follows that the leading term in ϵ of the second term in Eq. (B2) is of the order of $\epsilon^{\frac{k}{k+m}}$ for some $k,m \geqslant 1$. Hence, for ϵ small enough, the extra positive term of order $\epsilon^{\frac{k}{k+m}}$ in Eq. (B2) offsets the negative term of order ϵ^p in Eq. (B1), and we get, for some $c > 0$,

$$\mathcal{R}_p(\tilde{x}) = \left[1 - \frac{|I|}{p}\epsilon^p + o(\epsilon^p)\right]\left[\mathcal{R}_p(x) + c\epsilon^{\frac{k}{k+m}} + o(\epsilon^{\frac{k}{k+m}})\right]$$
$$= \mathcal{R}_p(x) + c\epsilon^{\frac{k}{k+m}} + o(\epsilon^{\frac{k}{k+m}}) > \mathcal{R}_p(x).$$

Having established that a maximizer x must be positive, $x > 0$, the remainder of the proof is the same as the proof for irreducible hypergraphs, since in Eq. (6) it suffices that at least one $j_m \notin I$ arrives at a contradiction, which is guaranteed by the connectedness of $\mathcal{H}$.

For directed hypergraphs, the condition (and definition) of strong connectivity in Appendix A is tailormade to ensure that the above argument still goes through. More precisely, if (x,y) are a pair of non-negative maximizers of $\mathcal{R}_{p,q}(x,y)$ [cf. Eq. (7)], then define $I = \{i \in \mathcal{V} : x_i = 0\}$ and $J = \{j \in \mathcal{V} : y_j = 0\}$. Setting the zero elements in x and y to a small positive value ϵ, strong connectivity implies that the numerator of $\mathcal{R}_{p,q}$ increases by a term of order ϵ^α with $\alpha < 1$, whereas the denominator (the norms of x and y) can only decrease $\mathcal{R}_{p,q}$ by a term of order $\epsilon^{\frac{p+q}{2}}$ with $p,q \geqslant 1$. The uniqueness argument again follows along the lines leading to Eq. (6).

APPENDIX C: NETWORK DATA AND NUMERICAL SETTINGS

Here we summarize the data sources and parameter settings used in the example applications (Secs. VII and VIII).

1. Random geometric graphs

A geometric graph with N vertices and radius r is defined by a set $\mathcal{V}$ of points in a metric space and edges $\mathcal{E} = \{(u,v) \in \mathcal{V} : 0 < \|u - v\| \leqslant r\}$. We generated random geometric graphs by sampling with uniform probability N points in the unit square $[0,1] \times [0,1]$ and taking the standard two-norm as the distance measure. For a given vertex, the probability that it is connected to any other vertex is πr^2. Hence, if we increase N while keeping $\rho = Nr^2$ constant, we obtain a sequence of random geometric graphs with constant average expected degree.

2. Alignment of yeast and human PPI networks

We obtained physical protein-protein interactions (PPI) for yeast from the BioGRID [58] database and physical and functional PPIs for human from the BioGRID and STRING [59] databases. The yeast network had 36 391 interactions between 4847 proteins; the human network had 40 630 interactions between 9602 proteins. We integrated these networks with orthology mappings from the InParanoid database [60]. There were 3390 orthology relations between 2245 yeast and 3255 human proteins which had at least one interaction in their respective PPI networks. We performed recursive spectral clustering on the directed alignment hypergraph consisting of 2567 interolog hyperedges [cf. Fig. 3(a)]. At $p = q = 1$, 180 clusters with at least two hyperedges were found; 119 hyperedges had no connections in the hypergraph, forming singleton clusters. The complete distribution of hyperedges, nodes, and scores for all clusters is shown in Fig. S2 of the Supplemental Material [38]; the functional analysis of the local and global alignment clusters is given in Tables S2 and S3 of the Supplemental Material [38].

3. Tripartite community detection in the CiteULike data

We obtained the complete "who-posted-what" data from CiteULike [48], containing (as of 1 February 2012) 16 553 642 (user, article, tag) entries. To create a more manageable data set, we considered all entries from 2005, resulting in a hypergraph of 466 948 (user, article, tag) hyperedges between 4693 users, 121 071 articles, and 36 489 tags. Recursive hypergraph spectral clustering with $p = 1$ identified 13 987 clusters with at least two hyperedges; 4616 hyperedges formed singleton clusters. The complete distribution of hyperedges, nodes, and scores for all clusters is shown in Fig. S3 of the Supplemental Material [38]. While comparing the user, article, and tag overlap of two hyperedge clusters, we were primarily interested to detect when the set of users, articles, or tags of a smaller cluster is entirely contained in a larger cluster (cf. Fig. 4). We therefore used the overlap score defined for two sets X and Y as

$$\text{ovlp}(X,Y) = \frac{|X \cap Y|}{\min(|X|,|Y|)},$$

which reaches its maximum value of 1 whenever $X \subset Y$ or $Y \subset X$.

4. Path clustering in the yeast transcriptional regulatory network

We obtained a network of 11 373 physical transcription factor (TF) binding interactions between 198 TFs and 3535 target genes in yeast from [50] and knock-out microarray data for 266 TFs from [49]. The knock-out data can be represented as a directed network of perturbational interactions where each TF is connected to the genes which respond to the knock-out perturbation of that TF. In addition, 182 TFs with physical binding data also had knock-out data for a total of 7090 perturbational interactions. We constructed a directed hypergraph consisting of 1332 hyperedges and 788 nodes, where each hyperedge is a shortest path in the regulatory network between a TF and a gene differentially expressed upon knock out of that TF. We defined the source set of a hyperedge as the knocked-out TF and the target set as the remainder of the path. Recursive spectral clustering identified 39 clusters of which 14 were singletons.

5. Supplementary data and algorithm implementation

An implementation of the clustering algorithm in JAVA, together with the input data and clustering results described in Sec. VIII, is available from the project home page in Ref. [61].

[1] M. E. J. Newman, SIAM Rev. **45**, 167 (2003).
[2] R. Albert and A-L. Barabási, Rev. Mod. Phys. **74**, 47 (2002).
[3] R. Milo, S. Shen-Orr, S. Itzkovitz, N. Kashtan, D. Chklovskii, and U. Alon, Science **298**, 824 (2002).
[4] M. E. J. Newman, Proc. Natl. Acad. Sci. USA **103**, 8577 (2006).
[5] S. Fortunato, Phys. Rep. **486**, 75 (2010).
[6] J. Leskovec, K. J. Lang, A. Dasgupta, and M. W. Mahoney, Internet Math. **6**, 29 (2009).
[7] M. A. Porter, J.-P. Onnela, and P. J. Mucha, Notices AMS **56**, 1082 (2009).
[8] G. Ghoshal, V. Zlatić, G. Caldarelli, and M. E. J. Newman, Phys. Rev. E **79**, 066118 (2009).
[9] V. Zlatić, G. Ghoshal, and G. Caldarelli, Phys. Rev. E **80**, 036118 (2009).
[10] U. Alon, *An Introduction to Systems Biology: Design Principles of Biological Circuits* (Chapman & Hall/CRC, London, 2007).
[11] X. Zhu, M. Gerstein, and M. Snyder, Genes Dev. **21**, 1010 (2007).
[12] R. Sharan and T. Ideker, Nature Biotechnol. **24**, 427 (2006).
[13] D. Zhou, J. Huang, and B. Schölkopf, in *Advances in Neural Information Processing Systems (NIPS) 19*, edited by B. Schölkopf, J. C. Platt, and T. Hofmann (MIT Press, Cambridge, MA, 2007), pp. 1601–1608.
[14] A. Vazquez, Phys. Rev. E **77**, 066106 (2008).
[15] S. Klamt, U-U. Haus, and F. Theis, PLoS Comp. Biol. **5**, e1000385 (2009).
[16] A. Vazquez, J. Stat. Mech.: Theory Exp. (2009) P07006.
[17] M. E. J. Newman, Phys. Rev. E **74**, 036104 (2006).
[18] R. A. Horn and C. R. Johnson, *Matrix Analysis* (Cambridge University Press, Cambridge, England, 1985).
[19] G. H. Golub and C. F. Van Loan, *Matrix Computations*, 3rd ed. (The Johns Hopkins University Press, Baltimore, MD, 1996).
[20] T. S. Evans and R. Lambiotte, Phys. Rev. E **80**, 016105 (2009).
[21] Y. Y. Ahn, J. P. Bagrow, and S. Lehmann, Nature (London) **466**, 761 (2010).
[22] L. De Lathauwer, B. De Moor, and J. Vandewalle, SIAM J. Matrix Anal. Appl. **21**, 1324 (2000).
[23] K. Inoue and K. Urahama, Pattern Recogn. Lett. **20**, 699 (1999).
[24] J. M. Kleinberg, J. ACM **46**, 604 (1999).
[25] L-H. Lim, in *Proceedings of the IEEE International Workshop on Computational Advances in Multi-Sensor Adaptive Processing* (IEEE, 2005), pp. 129–132.
[26] K. C. Chang, K. Pearson, and T. Zhang, Commun. Math. Sci. **6**, 507 (2008).
[27] S. Friedland, S. Gaubert, and L. Han, Lin. Alg. Appl., doi: 10.1016/j.laa.2011.02.042 (2011).
[28] T. Michoel, A. Joshi, B. Nachtergaele, and Y. Van de Peer, Mol. Bio. Syst. **7**, 2769 (2011).
[29] P. Audenaert, T. Van Parys, M. Pickavet, P. Demeester, Y. Van de Peer, and T. Michoel, Bioinformatics **27**, 1587 (2011).
[30] L. V. Zhang, O. D. King, S. L. Wong, D. S. Goldberg, A. H. Y. Tong, G. Lesage, B. Andrews, H. Bussey, C. Boone, and F. P. Roth, J. Biol. **4**, 6 (2005).
[31] Daniel M. Dunlavy, Tamara G. Kolda, and W. Philip Kegelmeyer, Tech. Rep. No. SAND2006-2079 (Sandia National Laboratories, Albuquerque, NM and Livermore, CA, 2006).
[32] W. Li, C-C. Lie, T. Zhang, H. Li, M. S. Waterman, and X. J. Zhou, PLoS Comp. Biol. **7**, e1001106 (2011).
[33] P. J. Mucha, T. Richardson, K. Macon, M. A. Porter, and J. P. Onnela, Science **328**, 876 (2010).
[34] S. Brin and L. Page, Comput. Networks **30**, 107 (1998).
[35] G. Palla, I. Derényi, I. Farkas, and T. Vicsek, Nature (London) **435**, 814 (2005).
[36] S. Fortunato and M. Barthélemy, Proc. Natl. Acad. Sci. USA **104**, 36 (2007).
[37] B. H. Good, Y-A. de Montjoye, and A. Clauset, Phys. Rev. E **81**, 046106 (2010).
[38] See Supplemental Material at http://link.aps.org/supplemental/10.1103/PhysRevE.86.056111 for supplementary figures and tables.
[39] O. Kuchaiev, T. Milenkovic, V. Memisevic, W. Hayes, and N. Przulj, J. R. Soc. Interface **7**, 1341 (2010).
[40] J. Berg and M. Lässig, Proc. Natl. Acad. Sci. USA **103**, 10967 (2006).
[41] R. Sharan, S. Suthram, R. M. Kelley, T. Kuhn, S. McCuine, P. Uetz, T. Sittler, R. M. Karp, and T. Ideker, Proc. Natl. Acad. Sci. USA **102**, 1974 (2005).
[42] H. Yu, N. M. Luscombe, H. X. Lu, X. Zhu, Y. Xia, J. D. J. Han, N. Bertin, S. Chung, M. Vidal, and M. Gerstein, Genome Res. **14**, 1107 (2004).
[43] B. P. Kelley, R. Sharan, R. M. Karp, T. Sittler, D. E. Root, B. R. Stockwell, and T. Ideker, Proc. Natl. Acad. Sci. USA **100**, 11394 (2003).
[44] A-L. Barabási and Z. N. Oltvai, Nature Rev. Genet. **5**, 101 (2004).
[45] B. K. Tye, Annu. Rev. Biochem. **68**, 649 (1999).
[46] H. Kibak, L. Taiz, T. Starke, P. Bernasconi, and J. P. Gogarten, J. Bioenerg. Biomemb. **24**, 415 (1992).
[47] http://www.flickr.com.
[48] http://www.citeulike.org/faq/data.adp.
[49] Z. Hu, P. J. Killion, and V. R. Iyer, Nature Genet. **39**, 683 (2007).
[50] C. T. Harbison, D. B. Gordon, T. I. Lee, N. J. Rinaldi, K. D. Macisaac, T. W. Danford, N. M. Hannett, J. B. Tagne, D. B. Reynolds, J. Yoo, E. G. Jennings, J. Zeitlinger, D. K. Pokholok, M. Kellis, P. A. Rolfe, K. T. Takusagawa, E. S. Lander, D. K. Gifford, E. Fraenkel, and R. A. Young, Nature (London) **431**, 99 (2004).
[51] T. Ideker, O. Ozier, B. Schwikowski, and A. F. Siegel, Bioinformatics **18**, S233 (2002).
[52] C. T. Workman, H. C. Mak, S. McCuine, J. B. Tagne, M. Agarwal, O. Ozier, T. J. Begley, L. D. Samson, and T. Ideker, Science **312**, 1054 (2006).
[53] A. Gitter, Z. Siegfried, M. Klutstein, O. Fornes, B. Oliva, I. Simon, and Z. Bar-Joseph, Mol. Syst. Biol. **5**, 276 (2009).
[54] A. Joshi, T. Van Parys, Y. Van de Peer, and T. Michoel, Genome Biol. **11**, R32 (2010).
[55] R. Guimerà, M. Sales-Pardo, and L. A. Nunes Amaral, Phys. Rev. E **76**, 036102 (2007).
[56] G. Palla, I. J. Farkas, P. Pollner, I. Derényi, and T. Vicsek, New J. Phys. **9**, 186 (2007).
[57] E. A. Leicht and M. E. J. Newman, Phys. Rev. Lett. **100**, 118703 (2008).
[58] C. Stark, B-J. Breitkreutz, T. Reguly, L. Boucher, A. Breitkreutz, and M. Tyers, Nucleic Acids Res. **34**, D535 (2006).
[59] L. J. Jensen, M. Kuhn, M. Stark, S. Chaffron, C. Creevey, J. Muller, T. Doerks, P. Julien, A. Roth, M. Simonovic *et al.*, Nucleic Acids Res. **37**, D412 (2009).
[60] A. C. Berglund, E. Sjolund, G. Ostlund, and E. L. L. Sonnhammer, Nucleic Acids Res. **36**, D263 (2008).
[61] http://schype.googlecode.com.