The Annals of Applied Statistics
2014, Vol. 8, No. 3, 1853–1891
DOI: 10.1214/14-AOAS760

A TESTING BASED EXTRACTION ALGORITHM FOR IDENTIFYING SIGNIFICANT COMMUNITIES IN NETWORKS[1]

BY JAMES D. WILSON, SIMI WANG, PETER J. MUCHA[2], SHANKAR BHAMIDI AND ANDREW B. NOBEL

University of North Carolina at Chapel Hill

A common and important problem arising in the study of networks is how to divide the vertices of a given network into one or more groups, called communities, in such a way that vertices of the same community are more interconnected than vertices belonging to different ones. We propose and investigate a testing based community detection procedure called Extraction of Statistically Significant Communities (ESSC). The ESSC procedure is based on p-values for the strength of connection between a single vertex and a set of vertices under a reference distribution derived from a conditional configuration network model. The procedure automatically selects both the number of communities in the network and their size. Moreover, ESSC can handle overlapping communities and, unlike the majority of existing methods, identifies "background" vertices that do not belong to a well-defined community. The method has only one parameter, which controls the stringency of the hypothesis tests. We investigate the performance and potential use of ESSC and compare it with a number of existing methods, through a validation study using four real network data sets. In addition, we carry out a simulation study to assess the effectiveness of ESSC in networks with various types of community structure, including networks with overlapping communities and those with background vertices. These results suggest that ESSC is an effective exploratory tool for the discovery of relevant community structure in complex network systems. Data and software are available at http://www.unc.edu/~jameswd/research.html.

1. Introduction. The study of networks has been motivated by, and made significant contributions to, the modeling and understanding of complex systems. Networks are used to model the relational structure between individual units of an observed system. In the network setting, vertices represent the units of the system and edges are placed between vertices that are related in some way. Network-based models have been used in a variety of disciplines: in biology to model protein-protein and gene–gene interactions; in sociology to model friendship and infor-

Received September 2013; revised May 2014.

[1]Supported in part by NSF Grants DMS-09-07177, DMS-13-10002, DMS-06-45369, DMS-11-05581 and SES-1357622.

[2]Supported in part by the James S. McDonnell Foundation 21st Century Science Initiative—Complex Systems Scholar Award Grant 220020315.

Key words and phrases. Community detection, networks, extraction, background, multiple testing.

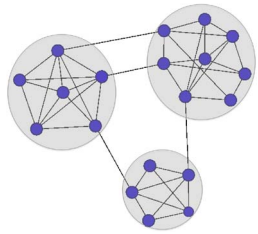

FIG. 1. *A simple network with three distinct communities.*

mation flow among a group of individuals; and in neuroscience to model the relationship between the organization and function of the brain. In many of these applications, the vertices of the network under study can naturally be subdivided into communities. Informally, a community is a group of vertices that are more connected to each other than they are to the remainder of the network. More rigorous definitions quantify this notion of differential connection in different ways. Figure 1 illustrates a network with three disjoint communities.

The problem of dividing the vertices of a given network into well-defined communities is known as community detection. Community detection has become increasingly popular, as communities have been found to identify important and useful features of many complex systems. Community detection has been studied by researchers in a variety of fields, including statistics, the social sciences, computer science, physics and applied mathematics, and a diverse set of community detection algorithms have been developed [see Fortunato (2010), Porter, Onnela and Mucha (2009) for reviews].

Existing community detection methods capture different types of community structure. The simplest community structure, and the one most commonly studied, is a hard partitioning, in which each vertex of the network is assigned to one and only one community, and the collection of communities together form a partition of the network [e.g., Newman and Girvan (2004), Ng, Jordan and Weiss (2002), Snijders and Nowicki (1997)]. Another class of community structure allows overlapping communities [see Xie, Kelley and Szymanski (2011) for a recent review], in which the collection of communities together form a cover of the network. Broadly speaking, most community detection methods produce one of these types of structures.

Community detection has been successful in understanding a wide variety of complex systems. In addition to the numerous examples cited in the aforementioned reviews, community detection methods have recently been profitably applied to protein interaction networks [Lewis et al. (2010)], functional brain activity [Bassett et al. (2011)], social media [Papadopoulos et al. (2012)] and mobile phone data [Muhammad and Van Laerhoven (2013)], as well as social groups [Greene, Doyle and Cunningham (2010), Miritello, Moro and Lara (2011), Onnela et al. (2011)].

The majority of existing community detection methods make the assumption that *every* vertex within an observed network belongs to at least one community. Though many networks can be appropriately divided into a partition (or cover) of communities, some large and heterogeneous networks do not fit into this framework. For example, consider the Enron email network from Leskovec et al. (2009) where edges represent the email correspondence (sent or received) between email accounts in 2001. The network contains many (on the order of 10K) email accounts outside of Enron and relatively few (on the order of 1K) email accounts from employees at Enron. The outside email accounts, many of which are spam email accounts, are not preferentially attached to any group of employees and thereby do not belong to a well-defined community. From this example and several others that we investigate in Section 4, we will see that many real networks contain vertices that do not have strong connections to *any* community. Informally, we call vertices that are not preferentially connected to any community *background* vertices, as they act as a background against which more standard community structures may be detected.

In networks where background vertices are present, partitioning and covering methods typically assign them to more tightly connected communities. To illustrate this, we generated a 500 node toy network with a single community of size 50, whose vertices are linked independently with probability 0.5; the remaining vertices are background and are linked to all vertices in the network independently with probability 0.05. We ran two popular detection methods—the modularity based algorithm of Newman and Girvan (2004) and the normalized Spectral algorithm of Ng, Jordan and Weiss (2002)—and found two disjoint communities. We considered the community that most closely matched the true embedded community and found, as shown in Figure 2, that both methods included many background vertices.

Also shown in Figure 2 is the result of applying the ESSC method introduced in this paper. ESSC accurately identifies the embedded community and the background, and separates one from the other. Although there are methods in multivariate clustering to capture background [Ester et al. (1996), Hinneburg and Keim (1998)], only a few recent papers, for example, Lancichinetti et al. (2011), Zhao, Levina and Zhu (2011), consider background in the context of community detection.

In this paper we propose and study a testing based community detection algorithm, called Extraction of Statistically Significant Communities (ESSC), that is capable of identifying both background vertices and overlapping communities. The core of the algorithm is an iterative search procedure that identifies statistically stable communities. In particular, the search procedure uses tail probabilities derived from a stochastic configuration model based on the observed network in order to assess the strength of the connection between a single vertex and a candidate community. Updating of the candidate community is carried out using ideas from multiple testing and false discovery rate control.

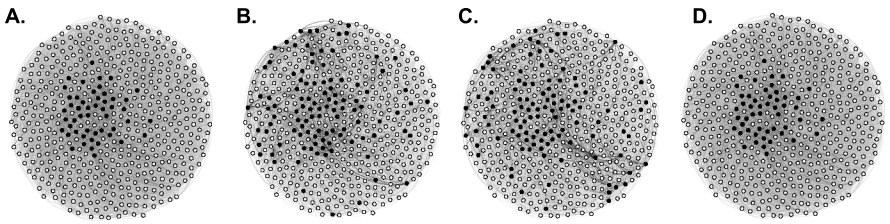

FIG. 2. (A) *A toy network that contains one significantly connected community—colored in black—and many sparsely connected background vertices*. (B) *The partition given by the GenLouvain modularity optimization method*. (C) *The partition given by normalized Spectral clustering*. (D) *The extracted community found by the proposed method ESSC, which separates and distinguishes the embedded community from the background.*

The only free parameter in the ESSC algorithm is a false discovery rate threshold that is used in the update step of the iterative search procedure. The number of detected communities, their overlap (if any) and the size of the background are handled automatically, without user input. In practice, the output of ESSC is not overly sensitive to the threshold parameter; see the Appendix D for more details.

1.1. *Notation*. For ease of discussion throughout the remainder of this paper, we first introduce some notation. Let $G = (V, E)$ be an undirected multigraph with vertex set $V = [n] = \{1, \ldots, n\}$ and edge multiset E containing all (unordered) pairs $\{i, j\}$ such that there is an edge between vertices i and j in G, allowing repetitions for multiple edges. Let $d(u)$ denote the degree of a vertex u, and let $\mathbf{d} = \{d(1), \ldots, d(n)\}$ denote the degree sequence of G. Let $B \subset [n]$ denote a subset of vertices in G. Indices on B are simply used for specification throughout. Write Π for a partition of the vertex set $[n]$ ($\Pi = B_1 \cup B_2 \cup \cdots \cup B_k$, $k \geq 1$). In many cases, detection methods seek a partition (or cover) through optimizing a specified quality or score function, which we will denote as $S(\cdot)$. It is important to note that the score may be global, in which case $S(\cdot)$ measures the quality of an entire partition, or local, in which case $S(\cdot)$ measures the quality of a potential community. We will use G_o to denote an observed graph and $\widehat{G}$ for a stochastic model on the vertex set $[n]$.

1.2. *Related work*. There is an extensive literature on the development and analysis of community detection methods. In this section we give an overview of this literature. For recent surveys describing community detection methods, see Fortunato (2010), Porter, Onnela and Mucha (2009) or Goldenberg et al. (2010). In Section 3 we describe in more detail the methods to which we compare ESSC.

Many of the earliest community detection methods approach network clustering from a graph-theoretic standpoint. Relying on a prespecified integer k, these methods seek the partition of k communities that minimize the number of edges

between communities. The optimal partition of this criterion is known as the partition of min-cut and max-flow [Goldberg and Tarjan (1988)], where the *cut* of a community specifies the number of edges from the community to the rest of the network. Unfortunately, min-cut methods often result in many singleton communities. To address this issue, the cut of a community can be normalized by either the community size, resulting in the ratio-cut criterion [Wei and Cheng (1989)], or by the total degree of the community, giving the normalized-cut criterion [Shi and Malik (2000)]. When $k > 2$, the task of finding the partition that satisfies any of these cut criterions is NP-hard. Spectral clustering methods [Krzakala et al. (2013), Ng, Jordan and Weiss (2002)] find an approximate solution to the norm-cut criterion by appealing to spectral properties of the graph Laplacian. Spectral clustering methods can be applied to either nonnetwork multivariate data or directly to relational network data.

Another class of community detection methods seek community structure by comparing the observed network $G_o = ([n], E_o)$ with an unstructured stochastic network on the same vertex set $\widehat{G}_{\text{null}} = ([n], \hat{E}_{\text{null}})$. A stochastic network $\widehat{G}_{\text{null}}$ describes the probabilities of edge connection between all pairs of vertices in $[n]$ given that each pair was connected at random. Detection methods of this class seek the partition of G_o whose clustering most deviates from what is expected under $\widehat{G}_{\text{null}}$. Modularity methods [see, e.g., Blondel et al. (2008), Clauset, Newman and Moore (2004), Mucha et al. (2010), Newman (2006)] are a popular subset of this class. Modularity methods seek the partition whose communities' fraction of observed edges are furthest from the fraction of edges expected under $\widehat{G}_{\text{null}}$, that is, the partition Π that maximizes

$$S_{\text{mod}}(\Pi) = \frac{1}{2|E_o|} \sum_{\ell=1}^{k} \Bigg(\sum_{i,j \in B_\ell} \mathbb{I}\big(\{i,j\} \in E_o\big) - \gamma \mathbb{E}\Bigg(\sum_{i,j \in B_\ell} \mathbb{I}\big(\{i,j\} \in \hat{E}_{\text{null}}\big)\Bigg)\Bigg),$$

where $\gamma > 0$ is a resolution parameter that controls the size of discovered communities. In many cases, γ is treated as one, however, this parameter can be tuned in a data-driven fashion. There are many choices for a reference stochastic network. For instance, in the case of the Newman–Girvan modularity [Newman and Girvan (2004)], $\widehat{G}_{\text{null}}$ is specified as the configuration model [Molloy and Reed (1995)] under which the degree sequence of G_o is maintained. In this case $\mathbb{E}(\sum_{i,j \in B_\ell} \mathbb{I}(\{i,j\} \in \hat{E}_{\text{null}}))$ is $d_o(i)d_o(j)/2|E_o|$. Our proposed method ESSC also relies upon the configuration model as a reference stochastic network.

An alternative class of community detection methods estimate the community structure of a network by fitting a structured stochastic network $\widehat{G}_{\text{struct}} = ([n], \hat{E}_{\text{struct}})$ to the observed data G_o. Here, $\widehat{G}_{\text{struct}}$ describes random assignments of edges conditional on stochastic community (or block) structure on the vertex set $[n]$. Formally, $\widehat{G}_{\text{struct}}$ is a parametric model whose parameters describe the community labels of each vertex and potentially the topological properties of the network (e.g., the degree distribution of the network). Given an observed network G_o

and a prespecified integer k, a structured network (with parameters Θ) is fit to G_o by maximizing the likelihood function describing Θ: $\mathcal{L}(\Theta|G_o, k)$. A recent review of structured network models is provided by Goldenberg et al. (2010). One of the most popular network models of this type is the stochastic block model [Holland, Laskey and Leinhardt (1983), Nowicki and Snijders (2001), Snijders and Nowicki (1997)]. Under this model, vertices are assigned labels taking values in $\{1, \ldots, k\}$ according to probabilities $\pi = (\pi_1, \ldots, \pi_k)$. Conditional on the vertex labels, edge probabilities are given by a $k \times k$ symmetric matrix $\mathbf{P}$ where the i, jth entry of $\mathbf{P}$ gives the probability of an edge between community i and j. Block models are fit to G_o by maximizing the corresponding likelihood $\mathcal{L}(\Theta = (\mathbf{P}, \pi)|G_o, k)$. Other examples of structured stochastic networks include latent variable models [Handcock, Raftery and Tantrum (2007), Hoff, Raftery and Handcock (2002)] and mixed membership models which are flexible to overlapping communities [Airoldi et al. (2008), Ball, Karrer and Newman (2011)].

Recently, there has been significant progress in the development of fast and efficient algorithms for fitting stochastic block models. The authors of Decelle et al. (2011) describe an algorithm that estimates block structure of a degree-corrected block model in time linear in the number of vertices. Their algorithm is based on a powerful heuristic of belief propagation from statistical physics. See, for example, Mézard and Montanari (2009) for a survey level treatment of belief propagation and a variety of applications. In the context of sparse stochastic block models, these techniques have been shown to be near optimal in estimating the underlying communities [Krzakala et al. (2013)], at least in the balanced regime where both communities are of equal size. A sublinear algorithm based on the pseudo-likelihood of the sparse block model is described in Amini et al. (2013) wherein block labels are shown to be consistent in the size of network. Finally, recent nonparametric representations of the block model through dense graph limits, or graphons [Airoldi, Costa and Chan (2013)] and network histograms [Olhede and Wolfe (2013)] provide promising new directions for the understanding and estimation of block models.

Another subclass of community detection methods are the so-called extraction techniques where communities are extracted one at a time [Lancichinetti et al. (2011), Zhao, Levina and Zhu (2011)]. Rather than search for an optimal partition or cover, these extraction methods seek the strongest connected community sequentially. Extraction methods do not force all vertices to be placed in a community and thereby are flexible to loosely connected background vertices. ESSC is an extraction method that utilizes the reference distribution of the connectivity of a community based on the conditional configuration model.

There are two main approaches currently used to assess the statistical significance of communities in networks. The first approach, like ESSC, builds upon statistical principles based on features of the observed network itself. The second approach is permutation based in that the significance of community structure is determined based on the results of a prescribed method on many bootstrapped

samples of the observed network [see, e.g., Clauset, Moore and Newman (2008), Rosvall and Bergstrom (2010)]. Many theoretical questions remain open for these types of methods, including convergence of bootstrapped samples of networks.

1.3. *Organization of the paper.* The remainder of this paper is organized as follows. Section 2 is devoted to a detailed description of our proposed algorithm for extraction of statistically significant communities (ESSC), including motivation and a description of the reference distribution generated from the configuration model. In Section 1.2 we discuss the competing methods that we use to validate our algorithm in both numerical and real network studies. In Section 4 we apply the ESSC algorithm to four real-world networks. These results provide solid evidence that ESSC performs well in practice, is competitive with (and in some cases arguably superior to) several leading community detection methods, and is effective in capturing background vertices. In Section 5 we propose a test bed of benchmark networks for assessing the performance of detection methods specifically on networks with background vertices. To the best of our knowledge, this is the first set of benchmarks proposed for networks of this type. We show that ESSC outperforms existing methods on these background benchmarks. We also show that ESSC performs competitively on standard (nonbackground) benchmark networks with both nonoverlapping and overlapping community structures. We end with a discussion of our work and avenues for future research.

2. The ESSC algorithm.

2.1. *Conditional configuration model.* Let G_o be an observed, undirected network having n vertices. Though many networks of interest will be simple, G_o may contain self-loops or multiple edges. Assume without loss of generality that G_o has vertex set $V = [n] = \{1, 2, \ldots, n\}$. The edge multiset E_o of G_o contains all (unordered) pairs $\{i, j\}$ such that $i, j \in [n]$ and there is a link between vertices i and j in G_o, with repetitions for multiple edges. Let $d_o(u)$ denote the degree of a vertex u, that is, the number of edges incident on u, and let $\mathbf{d}_o = \{d_o(1), \ldots, d_o(n)\}$ denote the degree sequence of G_o.

The starting point for our analysis is a stochastic network model that is derived from the degree sequence $\mathbf{d}_o$ of G_o, specifically, the configuration model associated with $\mathbf{d}_o$, which we denote by $\mathrm{CM}(\mathbf{d}_o)$ [Bender and Canfield (1978), Bollobás (1979), Molloy and Reed (1995)]. The configuration model $\mathrm{CM}(\mathbf{d}_o)$ is a probability measure on the family of multigraphs with vertex set $[n]$ and degree sequence $\mathbf{d}_o$ that reflects, within the constraints of the degree sequence, a random assignment of edges between vertices.

The configuration model $\mathrm{CM}(\mathbf{d}_o)$ has a simple generative form. Initially, each vertex $u \in [n]$ is assigned $d_o(u)$ "stubs," which act as half-edges. At the next stage, two stubs are chosen uniformly at random and connected to form an edge; this procedure is repeated independently until all stubs have been connected. Let $\widehat{G} =$

$([n], \hat{E})$ denote the random network generated by this procedure. Note that $\widehat{G}$ may contain self loops and multiple edges between vertices, even if the given network G is simple.

The configuration model $\mathrm{CM}(\mathbf{d}_o)$ is capable of capturing and preserving strongly heterogeneous degree distributions often encountered in real network data sets. Importantly, all edge probabilities in the configuration null model are determined solely by the degree sequence $\mathbf{d}_o$ of an observed graph. As a result, fitting a configuration model does not rely on simulation, rather, estimation only requires the degree sequence of a single observed graph.

Under the configuration model $\mathrm{CM}(\mathbf{d}_o)$ there are no preferential connections between vertices, beyond what is dictated by their degrees. As such, $\mathrm{CM}(\mathbf{d}_o)$ provides a reference measure against which we may assess the statistical significance of the connections between two sets of vertices in the observed network G_o: the more the observed number of cross-edges deviates from the expected number under the model, the greater the significance of the connection between the vertex sets. Let the observed network G_o and the random network $\widehat{G}$ be as above. Given a vertex $u \in [n]$ and vertex set $B \subseteq [n]$, let

$$d_o(u : B) = \sum_{v \in B} \sum_{e \in E_o} \mathbb{I}(e = \{u, v\})$$

denote the number of edges between u and some vertex in B in G_o. Define $\hat{d}(u : B)$ as the corresponding number of edges in $\widehat{G}$. Note that $\hat{d}(u : B)$ is a random variable taking values in the set $\{0, 1, \ldots, d_o(u)\}$, and that $d_o(u : B) = \hat{d}(u : B) = d_o(u)$ when $B = [n]$ is the full vertex set. We now state a theorem describing asymptotics for the random variable $\hat{d}(u : B)$ in the configuration model which will form the basis of the algorithm. Recall that the total variation distance between two probability mass functions $\mathbf{p} := \{p(i)\}_{i \geq 0}$ and $\mathbf{q} := \{q(i)\}_{i \geq 0}$ on the space of natural numbers $\mathbb{N}$ is defined by

$$d_{\mathrm{TV}}(\mathbf{p}, \mathbf{q}) := \frac{1}{2} \sum_{i=0}^{\infty} |p(i) - q(i)|.$$

THEOREM 1. *Let $\{\mathbf{d}_{o,n}\}_{n \geq 1}$ be the degree sequences of an observed sequence of graphs $\{G_o^n\}_{n \geq 1}$, where G_o^n is a graph with vertex set $[n]$ and edge set $E_{o,n}$. Let $\{\widehat{G}^n\}_{n \geq 1}$ be the corresponding random graphs on $[n]$ constructed via the configuration model. Let F_n be the empirical distribution of $\mathbf{d}_{o,n}$. Assume that there exists a cumulative distribution function F on $[0, \infty)$ with $0 < \mu := \int_{\mathbb{R}^+} x \, dF(x) < \infty$ such that*

$$F_n \xrightarrow{w} F \tag{2.1}$$

and

$$\int_{\mathbb{R}^+} x \, dF_n(x) \to \mu. \tag{2.2}$$

Fix $k \geq 1$. For each $n \geq 1$, let $u = u_n \in [n]$ be a vertex with degree $d_{o,n}(u) = k$ and let $B = B(n) \subseteq [n]$ be a set of vertices. Then the random variable $\hat{d}_n(u : B)$ is approximately Binomial$(k, p_n(B))$ *in the sense that*

$$d_{\mathrm{TV}}\big(\hat{d}_n(u : B), \mathrm{Bin}\big(k, p_n(B)\big)\big) \to 0,$$

as $n \to \infty$. Here

$$p_n(B) = \frac{\sum_{v \in B} d_{o,n}(v)}{\sum_{w \in [n]} d_{o,n}(w)} = \frac{1}{2|E_{o,n}|} \sum_{v \in B} d_{o,n}(v), \tag{2.3}$$

where $|E_{o,n}|$ is the total number of edges in the graph.

A precise proof of this fact is given in the Appendix A. In light of the fact that the configuration model CM($\mathbf{d}_o$) does not contain preferential connections between vertices, the probabilities

$$p(u : B) = P\big(\hat{d}(u : B) \geq d_o(u : B)\big) \tag{2.4}$$

can be used to assess the strength of connection between a vertex u and a set of vertices $B \subseteq [n]$. In particular, small values of $p(u : B)$ indicate that there are more edges between u and B than expected under the configuration model.

If we regard $d_o(u : B)$ as the observed value of a test statistic that is distributed as $\hat{d}(u : B)$ under the null model CM($\mathbf{d}_o$), then $p(u : B)$ has the form of a p-value for testing the hypothesis that u is not strongly associated with B.

This testing interpretation of $p(u : B)$ plays a role in the iterative search procedure that underlies the ESSC method (see below). However, we note that the testing point of view is informal, as the null model CM($\mathbf{d}_o$) itself depends on the observed network G_o through its degree distribution.

In general, the exact value of the probability $p(u : B)$ in (2.4) may be difficult to obtain. In practice, the ESSC procedure approximates $p(u : B)$ by $P(X_B \geq d_o(u : B))$, where X_B has a Binomial$(d(u), p(B))$ distribution appealing to the result of Theorem 1.

2.2. *Description of the ESSC algorithm.* The core of the ESSC algorithm is an iterative deterministic procedure (*Community-Search*) that searches for robust, statistically significant communities. Beginning with an initial set B_0 of vertices that acts as a seed, the procedure successively refines and updates B_0 using (the binomial approximation of) the probabilities (2.4) until it reaches a fixed point, that is, a vertex set that is unchanged under updating. The final vertex set identified by the search procedure is a detected community.

The *Community-Search* procedure is applied repeatedly, using an adaptively chosen sequence of seed vertices, until it returns an empty community with no nodes. The resulting collection $\mathcal{C}$ of detected communities (omitting repetitions) constitutes the output of the algorithm. The seed set B_0 for the initial run of the

search procedure is the vertex of highest degree and all of the vertices adjacent to it. In subsequent runs of the search procedure the seed set B_0 is the vertex of highest degree not contained in any previously detected community and all the vertices adjacent to it, regardless of whether the latter lie in a previously detected community or not.

To simplify what follows, let $C_1, \ldots, C_K$ be the distinct detected communities of G_o in $\mathcal{C}$. The background of G_o is defined to be the set of vertices that do not belong to any detected communities:

$$C_* = \text{Background}(G_o : \mathcal{C}) = [n] \Big\backslash \bigcup_{k=1}^{K} C_k. \tag{2.5}$$

In principle, the number K of detected communities can range from zero to n. Importantly, K is not fixed in advance, but is adaptively determined by the ESSC algorithm. The identification of detected communities by the *Community-Search* procedure allows communities to overlap. As with the number of discovered communities, K, the presence and extent of overlap is automatic; no prior specification of overlap specific parameters are required.

The updates of the *Community-Search* procedure bear further discussion. Consider an ideal setting in which, for each vertex u and vertex set B we can determine, in an unambiguous way, whether or not u is strongly connected to B in G_o. Informally, a set of vertices B is a community if the vertices $u \in B$ have a strong connection with vertices in B, while the vertices $u \in B^c$ do not. Equivalently, B is a community if and only if it is a fixed point of the update rule

$$S(A) = \{u \in [n] \text{ such that } u \text{ is strongly connected with } A\}$$

that identifies the vertices having a strong connection with a set of vertices $A \subseteq [n]$. Formally, we may regard $S(\cdot)$ as a map from the power set of $[n]$ to itself. A vertex set B is a fixed point of $S(\cdot)$ if $S(B) = B$. In order to find a fixed point of the update rule $S(\cdot)$, we apply the rule repeatedly, starting from a seed set of vertices B_0, until a fixed point is obtained. The eventual termination (and success) of this simple procedure is assured, as the power set of $[n]$ is finite. By the exhaustive or selective considering of appropriate seed sets we can effectively explore the space of fixed points of $S(\cdot)$, and thereby identify communities in G_o.

The choice of a seed set B_o for the *Community-Search* procedure requires further discussion. As currently implemented, we choose B_o as the neighborhood of the highest degree vertex among the vertices lying outside currently extracted communities. Consider the following situation, as pointed out by a referee, where one has two disconnected clusters C, C' such that C contains no inherent community structure, for example, an Erdős–Rényi random graph, and C' contains strong community structure, for example, a well-differentiated stochastic block model. If the maximal degree of C is larger than C', then ESSC could fail to find the community structure in C'. To address the above situation, one can run the *Community-Search* procedure in parallel across all vertex neighborhoods. In this case, the final

communities are the collection of uniquely extracted vertex sets. We found that the situation above did not arise in any of the applications or simulations that we investigate in this paper.

In practice, we make use of the probabilities $\{p(u : B) : u \in [n]\}$ to measure the strength of the connection between $u \in [n]$ and B relative to the reference distribution CM($\mathbf{d}$). In particular, we regard $p(u : B)$ informally as a p-value for testing the null hypothesis H_u^B that u is not preferentially connected to B. Then the task of identifying the vertices u preferentially connected to B amounts to rejecting a subset of the hypotheses $\{\mathrm{H}_u^B : u \in [n]\}$. This is accomplished in steps 4 and 5 of the *Community-Search* procedure, where we make use of an adaptive method of Benjamini and Hochberg [Benjamini and Hochberg (1995)] to reject a subset of the hypotheses. The rejection method ensures that the expected number of falsely rejected hypotheses divided by the total number of rejected hypotheses (the so-called false discovery rate) is at most α [see Benjamini and Hochberg (1995) for more details]. A default false discovery rate threshold α of 5% is common in many applications, and we adopt this value here. Pseudo-code for the *Community-Search* procedure and ESSC algorithm is shown below.

Community-Search Procedure

Given: Graph $G_o = ([n], E_o)$; significance level $\alpha \in (0, 1)$.
Input: Seed set $B_0 \subseteq [n]$.
Initialize: $t := -1$, $B_{-1} = \varnothing$.
Loop (Update): Until $B_{t+1} = B_t$
1. $t := t + 1$.
2. Compute $p(u : B_t)$ for each $u \in [n]$.
3. Order the n vertices of G_o so that $p(u_1 : B_t) \le \cdots \le p(u_n : B_t)$.
4. Let $k \ge 0$ be the largest integer such that $p(u_k : B) \le (k/n)\alpha$.
5. Update $B_{t+1} := \{u_1, \ldots, u_k\}$.

Return: Fixed point community B_t.

ESSC Algorithm

Input: Graph $G_o = ([n], E_o)$; significance level $\alpha \in (0, 1)$.
Initialize: $V = [n]$, $\mathcal{C} := \varnothing$.
Loop:
Let $u \in V$ be the smallest (in case of ties) vertex with maximal degree.
Define seed set $B_0 := \{u\} \cup \{v \in [n] : \{u, v\} \in E_o\}$.
Obtain detected community $C :=$ *Community-Search*(B_0) from search procedure.
If $C \neq \varnothing$ then
Update $\mathcal{C} := \mathcal{C} \cup \{C\}$.

Update $V := V \setminus C$.
Repeat Loop.
Otherwise (if $C = \varnothing$), terminate the procedure.
Return: Family $\mathcal{C}$ of detected communities.

3. Competing methods. Here we describe the set of community detection methods that we use for validation and comparison with ESSC. We implement a variety of established detection methods all of which have publicly available code. We note that we do not compare ESSC with the recently developed fast block model algorithms from Decelle et al. (2011), Airoldi, Costa and Chan (2013) and Krzakala et al. (2013); such comparisons would be interesting for future work. The parameter settings for each algorithm are described in the Appendix C.

GenLouvain: The GenLouvain method of Jutla, Jeub and Mucha (2011/2012) is a modularity-based method that employs an agglomerative optimization algorithm to search for the partition that maximizes the score in (1.2). The algorithm is composed of two stages that are repeated iteratively until a local optimum is reached. In the first, each vertex is assigned to its own distinct community. Then for each vertex u (of community B_u), the neighbors of u are sequentially added to B_u if the addition results in a positive change in modularity. This procedure is repeated for all vertices in the network until no positive change in modularity is possible. In the second stage of the algorithm, the communities found in the first stage are treated as the new vertex set and passed back to the first stage of the algorithm where two communities are treated as neighboring if they share at least one edge between them. Throughout the remainder of this paper, we specify $\widehat{G}_{\text{null}}$ as the configuration model so that GenLouvain is set to optimize the Newman–Girvan modularity [Newman and Girvan (2004)]. As a result, the Louvain methods of Blondel et al. (2008) and GenLouvain can be used interchangeably (notably, however, the GenLouvain code does not exploit all possible efficiencies for this null model).

Infomap: The Infomap method of Rosvall and Bergstrom (2008) is a flow-based method that seeks the partition that optimally compresses the information of a random walk through the network. In particular, the optimal partition minimizes the quality function known as the Map Equation [Rosvall, Axelsson and Bergstrom (2009)], which measures the description length of the random walk. The method employs the same greedy search algorithm as Louvain [Blondel et al. (2008)], refining the results through simulated annealing.

Spectral: Given a prespecified integer k, the Spectral method of Ng, Jordan and Weiss (2002) seeks the partition that best separates the k smallest eigenvectors of the graph Laplacian. Specifically, the k smallest eigenvectors of the graph Laplacian are stacked to form the $n \times k$ eigenvector matrix X and k-means clustering

is applied to the normalized rows of X. Vertices are then assigned to communities according to the results of k-means. We note that there are proposed heuristics for choosing k. For example, the algorithm in Krzakala et al. (2013) does not require one to specify the number of communities in advance and uses the number of real eigenvalues outside a certain disk in the complex plane as a starting estimate. Throughout the manuscript, however, we choose k based on characteristics of the data investigated.

ZLZ: The method of Zhao, Levina and Zhu (2011), which we informally call ZLZ, is an extraction method that searches for communities one at a time based on a local graph-theoretic criterion. In each extraction, ZLZ employs the Tabu search algorithm [Glover (1989)] to find the community B that maximizes the difference of within-community edge density and outer edge density:

$$|B||B^c| \sum_{i,j\in[n]} \left(\frac{A_{i,j}\mathbb{I}(i \in B, j \in B)}{|B|^2} - \frac{A_{i,j}\mathbb{I}(i \in B, j \in B^c)}{|B||B^c|} \right), \tag{3.1}$$

where $|B|$ denotes the number of vertices in B and $A_{i,j}$ is the i,jth entry of the adjacency matrix associated with the observed graph. Once a community is extracted, the vertices of the community are removed from the network and the procedure is repeated until a prespecified number of disjoint communities are found. By following a similar technique described in Bickel and Chen (2009), the authors show that under a degree-corrected block model, the estimated labels resulting from maximizing (3.1) are consistent as the size of the network tends to infinity [see Zhao, Levina and Zhu (2012) for more details].

OSLOM: The OSLOM method [Lancichinetti et al. (2011)] is an inferential extraction method that compares the local connectivity of a community with what is expected under the configuration model. Given a fixed collection of vertices B, the method first calculates the probability of all external vertices having at least as many edges as it has shared with the collection. These probabilities are then resampled from the observed distribution. The order statistics of the resampled probabilities are used to decide which vertices should be added to B; a vertex is added whenever the cumulative distribution function of its order statistic falls below a preset threshold α. Vertices are iteratively added and taken away from B in a stepwise fashion according to the above procedure. This extraction procedure is run across a random set of initializing communities and the final set of communities are pruned based on a pairwise comparison of overlap.

There are a few similarities between ESSC and these described competing methods. For instance OSLOM and GenLouvain both specify the configuration model as a reference network model to which candidate communities are compared. Both ZLZ and OSLOM are extraction methods, like ESSC, that do not require all vertices to belong to a community. The ESSC method uses the parametric distribution that approximates local connectivity of vertices and a candidate community. Since

TABLE 1

A summary of the detection methods we consider in our simulation and application study. From left to right, we list the type of community structure that each method can handle and the parameters required as input for each algorithm. Listed free parameters include the following: k, the number of communities; α, the significance level; N, the number of iterations; and γ, a resolution parameter

	Community structure			Free parameters			
Method	**Disjoint**	**Overlapping**	**Background**	k	α	N	γ
ESSC	✓	✓	✓		✓		
OSLOM	✓	✓	✓		✓	✓	
ZLZ	✓		✓	✓		✓	
GenLouvain	✓						✓
Infomap	✓				✓	✓	
Spectral	✓			✓		✓	

the configuration model can be estimated using only the observed graph, the probabilities in (2.3) have a closed form which can be computed analytically. On the other hand, OSLOM relies upon a bootstrapped sample of networks for determining the significance of a community. Whereas both OSLOM and ESSC are based on inferential statistical techniques, Infomap, Spectral, ZLZ and GenLouvain use network summaries directly. Unlike several of these mentioned methods, ESSC requires no specification of the number of communities and only relies upon one parameter which guides the false discovery rate. We summarize the features of ESSC and these competing methods in Table 1.

4. Real network analysis study. Existing community detection methods differ widely in their underlying criteria, as well as the algorithms they use to identify communities that satisfy these criteria. As such, we assess the performance of ESSC by comparing it with several existing methods—OSLOM, ZLZ, GenLouvain, Infomap, Spectral and k-means—on both a collection of real-world networks as well as an extensive collection of simulation benchmarks.

We first applied ESSC to four real networks of various size and density: the Caltech Facebook network [Traud et al. (2011)], the political blog network [Adamic and Glance (2005)], the personal Facebook network of the first author and the Enron email network [Leskovec et al. (2009)]. We summarize the network structures in Table 2 and visualize them in Figure 3.

On the first two networks, we compare quantitative features of the communities of each method, including size, number of communities, extent of overlap and extent of background. Moreover, we evaluate the ability of each method to capture specific features of these two complex networks through a formal classification study. We describe the precise settings of all tuning parameters for each of the detection algorithms in the Appendix C. All methods were run on a 4 GB RAM, 2.8 GHz dual processor personal computer.

TABLE 2
Summary statistics of the four networks that we analyze

Network	Number of vertices	Number of edges
Caltech	762	16,651
Political blog	1222	16,714
Personal Facebook	561	8375
Enron email	36,691	293,307

4.1. *Caltech Facebook network.* The Caltech Facebook network of Traud et al. (2011) represents the friendship relations of a group of undergraduate students at the California Institute of Technology on a single day in September, 2005. An edge is present between two individuals if they are friends on Facebook. In addition to friendship relations, several demographic features are available for each student, including dormitory residence, college major, year of entry, high school and gender. A summary of these features is given in Table 3. This data set provides a natural benchmark for community detection methods due to the possible association of community structure with one or more demographic features. Previous studies have found that this network displays community structure closely matching the dormitory residence of the individuals [Traud et al. (2011)]. We illustrate the network according to residence in Figure 3(A).

4.1.1. *Quantitative comparison.* We first compare the communities detected by each method based on quantitative summaries of the communities themselves: the number and size of the communities; the overlap present; and the number of background vertices found. A summary of the findings is given in Table 4. ESSC took 1.584 seconds to run on this network.

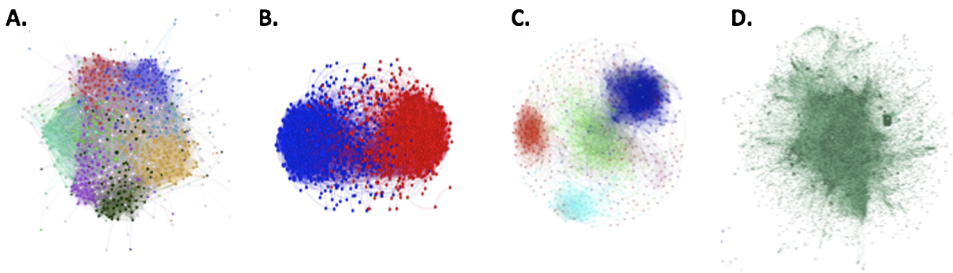

FIG. 3. *Real networks analyzed in the paper.* (A) *The Caltech Facebook network of* 2005 *colored by dormitory residence.* (B) *The* 2005 *political blog network colored by political affiliation.* (C) *The personal Facebook network of the first author colored by location in which he met each individual.* (D) *The Enron email network. Each graph is drawn with the Force Atlas* 2 *layout using Gephi software.*

TABLE 3

A summary of the features associated with the individuals in the Caltech Facebook network. From left to right, k is the number of unique categories, p_m is the proportion of missing data, m is the minimum size of any unique category, and M is the maximum size of any unique category

Feature	k	p_m	m	M
Dormitory	8	0.2205	44	98
Year	15	0.1457	1	173
Major	30	0.0984	1	88
High school	498	0.1693	1	3
Gender	2	0.0827	227	472

We note that the ZLZ, k-means and Spectral methods require prior specification of the number of discovered communities. Based on the ESSC and GenLouvain results, we ran each of these methods with seven and eight detected communities. We show the size distributions of the detected communities for each method in Figure 4, and find that the size distribution is broadly similar across the ESSC, ZLZ, GenLouvain and Spectral methods. Infomap found many ($N_C = 18$) small communities, including several communities of size three or fewer. At both $k = 7$ and 8, k-means found one large community as well as many small similarly sized communities. Interestingly, GenLouvain also produced an eighth community of

TABLE 4

*A summary of the detection methods run on the Caltech Facebook network. From left to right, N_C is the number of communities detected, $\overline{S}$ is the average size of the communities, $\hat{\sigma}_S$ is the standard deviation of the community size, $\overline{\mathbb{M}}$ is the average number of communities to which nonbackground vertices belong, $\overline{D}_{\text{sig}}$ is the average degree of the vertices in a community, $\overline{D}_B$ is the average degree of the background vertices, P_B is the proportion of background vertices, and $\hat{E}$ is the mean classification error associated with the dormitory feature of the individuals. *Methods were set to find 7 and 8 communities, based on the number of communities detected by ESSC and GenLouvain. —: represents repeated values*

Method	N_C	$\overline{S}$	$\hat{\sigma}_S$	$\overline{\mathbb{M}}$	$\overline{D}_{\text{sig}}$	$\overline{D}_B$	P_B	$\hat{E}$
ESSC	7	78.57	16.03	1.034	55.75	15.81	0.3018	0.0925
OSLOM	18	86.78	63.25	1.085	50.30	6.18	0.1496	0.2011
ZLZ*	7	62.14	41.97	1	64.08	16.60	0.4291	0.5346
ZLZ*	8	58	40.58	–	62.44	14.53	0.3911	0.5323
GenLouvain	8	95.25	35.75	–	43.70	NA	NA	0.2576
Infomap	18	42.33	46.23	–	–	–	–	0.8132
Spectral*	7	108.86	72.77	–	–	–	–	0.4865
Spectral*	8	95.25	61.52	–	–	–	–	0.4512
k-means*	7	108.86	126.51	–	–	–	–	0.4242
k-means*	8	95.25	118.35	–	–	–	–	0.4327

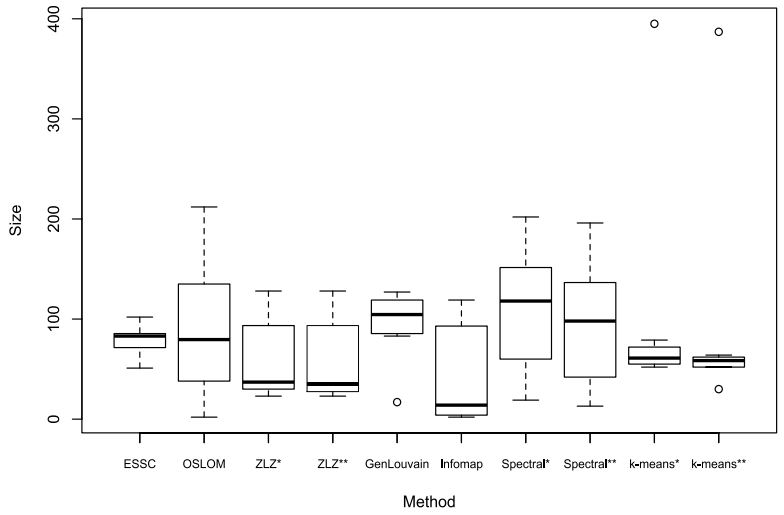

FIG. 4. *The size distributions of communities from each detection method when run on the Caltech network.*

size twenty-one, all of whose vertices were part of the background vertex set determined by ESSC. No method found significant overlap among the detected communities. The average number of communities to which each vertex belonged ranged from 1 to 1.085. Each of the methods capable of detecting background (ESSC, OSLOM and ZLZ) designated more than 15% of the total network as background, and vertices contained within communities had average degree nearly three times that of background vertices. This suggests, as expected, that the background vertices are less connected to other vertices in the network.

4.1.2. *Community features.* One motivation for community detection methods is their ability to find communities of vertices that represent interesting, but possibly unavailable, features of the system under study. Here, we explore the ability of each method to capture the demographic features of the Caltech network. To do this, we measure the extent to which the demographic features "cluster" within communities. Typical pair counting measures do not work well here, as the detected communities may overlap and may not cover the entire network. Also, pair counting measures treat the features as a "ground truth" partition of the network, whereas the true structure of a network is often more complex [Lee and Cunningham (2013), Yang and Leskovec (2012)]. As an alternative, we address the connection between communities and features through the problem of classification [see, e.g., Hastie, Tibshirani and Friedman (2001), Shabalin et al. (2009)]: for each vertex, we treat its community identification as a predictor and its demographic features as a discrete response that we wish to predict. We describe our approach in more detail.

Suppose that a detection method divides the vertices of the network into K communities plus background. Then the $n \times K$ matrix $X = [x_{i,j}]$ defined by

$$x_{i,j} = \begin{cases} 1, & \text{if vertex } i \text{ belongs to community } j, \\ 0, & \text{otherwise,} \end{cases}$$

represents the detected community structure of the network. For a given demographic feature α taking L-values, let $y_i^\alpha \in [L]$ be the value of α in sample i. We ignore samples for which the value of feature α is not available. Treating the ith row of the matrix X as a K-variate predictor for y_i^α, we use the Adaboost classification method [Freund and Schapire (1997)] with tree classifiers to construct a prediction rule $\phi : \{0, 1\}^K \to [L]$.

To evaluate each method, we first randomly divide the n samples into ten equally sized subgroups. Then by setting aside one subgroup as a test set, we train the classifier on the remaining subgroups and predict the features of the test set. By subsequently treating each subgroup as a test set in this way, we calculate the misclassification error associated with each test. We report the average misclassification error $\hat{E}$ for each method as a means of comparison and report the results in Table 4. The distribution of errors is shown in Figure 5. Values of $\hat{E}$ near zero suggest that the detected community structure captures the clustering of the selected feature. We consider the dormitory residence of the network, as this feature has been shown to be most representative of the community structure in past studies [Traud, Mucha and Porter (2012)]. From Figure 5, we see that ESSC has the lowest misclassification error among competing methods in this classification study.

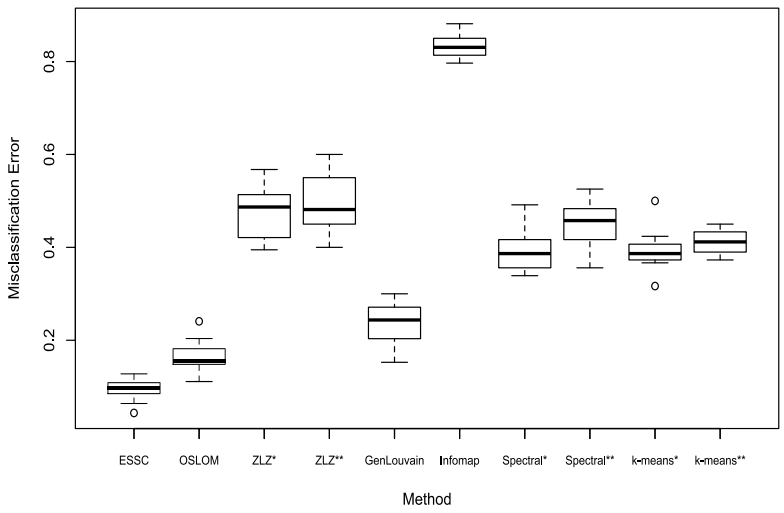

FIG. 5. *The misclassification error of each method based on the ten-fold classification study performed on the Caltech network. The community containment of each individual was used to classify his/her dormitory residence. For each test, an Adaboost classifier was used for comparison.*

These results suggest that the detected communities of ESSC best match the dormitory residence of the Caltech network.

4.2. *Political blog network.* The political blog network of Adamic and Glance (2005) represents the hyperlink structure of 1222 political blogs in 2005 near the time of the 2004 U.S. election. Undirected edges connect two blogs that have at least one hyperlink between them. The blogs were pre-classified according to political affiliation by the authors in Adamic and Glance (2005). These authors, as well as those of Newman (2006), observed that blogs of a similar political affiliation tend to link to one another much more often than to blogs of the opposite affiliation. We show a force directed layout of this network colored by political affiliation in Figure 3(B).

4.2.1. *Quantitative comparison.* We first compare the communities detected by each method based on their quantitative characteristics. The results are summarized in Table 5. ESSC took 2.012 seconds to run on this network.

Both the ESSC algorithm and GenLouvain found two large communities of similar size. Interestingly, Infomap found thirty-six communities, thirty-four of which contained fewer than 25 vertices. Roughly 95% of the vertices in these smaller communities of Infomap were contained in the background vertices of ESSC. Neither ESSC nor OSLOM found significant overlap among the communities, reflecting the tendency of the political bloggers to communicate with like-minded individuals: as noted by the authors of Adamic and Glance (2005), "divided they blog."

ESSC, OSLOM and ZLZ each assigned over twenty percent of the vertices to background. The pairwise Jaccard score of these background sets is greater than 0.67 in each case. The background vertices of all three extraction methods had mean degree six times smaller than vertices within communities, suggesting the presence of sparsely connected background vertices in this network.

TABLE 5

*A summary of the detection methods run on the Political blog network. The statistics shown here are the same as those in Table 4. *We set k to 2 to match the results of GenLouvain and ESSC. **We chose k as 10 so that at least 50 percent of the vertices were placed in a community*

Method	N_C	$\overline{S}$	$\hat{\sigma}_S$	$\overline{\mathbb{M}}$	$\overline{D}_{\text{sig}}$	$\overline{D}_B$	P_B	$\hat{E}$
ESSC	2	448.50	75.66	1	36.322	2.577	0.2651	0.0201
OSLOM	11	87.58	79.48	1.110	33.749	5.342	0.225	0.0306
ZLZ**	10	60.00	37.69	1	35.50	2.50	0.506	0.1341
GenLouvain	2	611.00	72.12	–	27.36	NA	0	0.0475
Infomap	36	33.94	125.74	–	–	–	–	0.0532
Spectral*	2	611.00	858.43	–	–	–	–	0.3821
k-means*	2	611.00	613.77	–	–	–	–	0.2856

4.2.2. *Political affiliation*. We now evaluate the extent to which the political affiliation of the blogs "cluster" by conducting the same classification study detailed in Section 4.1.2. We report the mean proportion of misclassified labels $\hat{E}$ in Table 5. ESSC, OSLOM, GenLouvain and Infomap all maintained classification errors below 10%, suggesting that political affiliation is captured by the network's community structure quite well. ESSC had the lowest misclassification error in this study, keeping an error below 4% across all tests. We look deeper into the strength of connection of the background vertices to the true political affiliations. Interestingly, these vertices were still preferentially attached to their true affiliation, however, their associated p-values were typically greater than 0.10, indicating weak affiliation.

4.3. *Personal Facebook network*. The personal Facebook network gives friendship structure of the first author's friends on Facebook. In addition, each individual is labeled according to the time period during which he or she met the first author. This data set, as well as the labels, is provided in the supplemental file [Wilson (2014)]. This network is shown, colored by label, in Figure 3(C).

The understanding of human social interactions has been improved through the analysis of large available social networks like Facebook [Lee and Cunningham (2013), Traud, Mucha and Porter (2012), Traud et al. (2011)]. Typically, these networks capture the social activity of individuals of a single location. For example, the Facebook network analyzed in Section 4.1 reflects the friendships of individuals specifically from the California Institute of Technology. The personal Facebook network provides one view of how individuals from different schools and locations interact given that they all have one friend in common.

We ran ESSC on the network (running time about 1 second) and found 7 communities with sizes varying from 10 to 157; see Table 6. Approximately 18% of the nodes in the network were distinguished as background. The mean degree of the vertices belonging to a community ($\overline{D}_{\text{sig}} \approx 33$) was about seven times that of the background ($\overline{D}_B \approx 5$). Of the vertices that were contained in a community, the average membership was very close to 1, suggesting little overlap between communities.

To understand how the location feature of the individuals cluster, we investigate the composition of each label according to detected community in Figure 6 and find several interesting results. The individuals from locations A, B, C, D and G all tend to cluster according to the detected communities. For instance, 79% of the individuals from location A were contained in community 5. Similarly, 60% or more of the individuals from locations B, C, D and G also belong to a single community in each case. Groups A, B, C and D represent the schools that the author attended from high school to final graduate school and make up nearly 81% of the total network. Groups E and F are not captured well by the communities, however, this is expected due to the small size of these locations ($n = 3$ in both cases). Finally, the most highly represented group among the background distinguished by

TABLE 6

Features of the personal Facebook network as well as the results of ESSC. On the left, we list the labels of the individuals according to location and the size of each group. On the right, we list the detected communities and background as well as their corresponding size

True features		**ESSC results**	
Label	**Size**	**Community**	**Size**
Acquaintance	80	1	43
A	62	2	107
B	94	3	75
C	150	4	157
D	147	5	53
E	3	6	26
F	3	7	10
G	22	Background	101 (18.0%)

ESSC were acquaintances—individuals met through other friends, events or conferences. These results suggest that friendships in this network cluster are based on location and that the acquaintances of the author are not well connected to his remaining friends.

4.4. *Enron email network.* The Enron email network from Leskovec et al. (2009) is a large (36,691 vertices), sparse network in which each vertex represents a unique email address. An undirected edge connects any two addresses if at least one email message has been sent from one address to the other. At least one vertex of each edge corresponds to the email address of an employee of the Enron

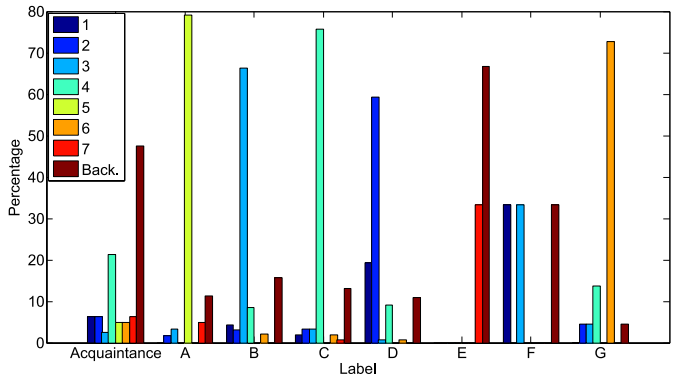

FIG. 6. *A bar plot showing the clustering of locations A–G and Acquaintances of the personal Facebook network. For each location label, we show the percentage of individuals from that location that were contained in each detected community. Communities are labeled* 1–7 *and Back. represents the background vertices.*

corporation. The network is shown in Figure 3(D). We ran ESSC on the network with $\alpha = 0.05$. ESSC took approximately 10 minutes to run on this network.

Importantly, the network includes Enron employees as well as advertising agencies and spam sites outside Enron. As such, we expect there to be many background vertices representing spam and advertisement email addresses. On applying ESSC to the network, we indeed find an abundance of background vertices—nearly 83% (30,454 vertices) of the network. The average degree of the vertices within a community is nearly twelve times that of the background vertices. ESSC found 8 communities with average size of 1239 and standard deviation 450. The average membership of the vertices that were contained within a community was 1.409, indicating a moderate amount of overlap of communities.

5. Simulation study. In this section we evaluate the performance of ESSC on simulated networks with three primary types of community structure: (1) communities that partition the network; (2) communities that overlap and cover the network; and (3) disjoint communities plus background.

Networks of the first two types have been well studied, and there are several existing simulation benchmarks for these structures [Girvan and Newman (2002), Lancichinetti and Fortunato (2009a, 2009b)]. We make use of the Lancichinetti, Fortunato and Radicchi (LFR) benchmark from Lancichinetti and Fortunato (2009a, 2009b) in order to assess the performance of ESSC and other methods on networks of the first two types. Our principal reason for using the LFR simulation benchmark is its flexibility, as well as the fact that the power-law degree distribution it employs is representative of the degree of heterogeneity present in many real networks [Barabási and Albert (1999)]. ESSC performs well on these standard nonoverlapping and overlapping benchmarks, and is in fact competitive with the other detection methods in these settings. We evaluate the results on these benchmarks in the Appendix B.

Relatively little attention has been paid to networks with background vertices, and we are not aware of a simulation benchmark for networks of this sort. We therefore propose a flexible simulation benchmark for networks with background that extends the LFR benchmark, and use it to compare ESSC with competing methods.

In the remainder of the section, we first describe the LFR benchmarks of Lancichinetti and Fortunato (2009a, 2009b) and then show how these benchmarks can be extended to networks with background. We assess the performance of ESSC and other competing methods on networks with background using our proposed benchmark.

5.1. *The LFR benchmark.* The LFR benchmarks of Lancichinetti and Fortunato (2009a, 2009b) include a number of parameters that govern the community structure of the simulated network; a list is given in Table 7. The edge density of the simulated network is controlled through the size n of the network and the mean

TABLE 7
Description of the free parameters available with the LFR benchmark networks

Parameter	Description
n	Size of the network
$\mu \in (0, 1)$	Mixing parameter: the proportion of external community degree for each vertex
τ_1	Power-law exponent for degree distribution of network
τ_2	Power-law exponent for size distribution of communities in network
$\overline{D}$	Mean degree
$[s_1, s_2]$	Size range of each community: $s_1 =$ lower limit $s_2 =$ upper limit
$\rho \in (0, 1)$	Proportion of vertices contained in two communities (used in overlapping benchmark only)

degree $\overline{D}$. For example, sparse networks are represented by benchmarks with large n and small $\overline{D}$. The degree distribution of simulated networks follows a power law with exponent τ_1. Lower and upper limits of the degree distribution are set to maintain an average degree $\overline{D}$ among vertices in the network. The distribution of community sizes in the LFR benchmark follows a power law with exponent τ_2. The size range $[s_1, s_2]$ sets lower and upper limits on the size of communities in the network. Consider a vertex u and its community C. Then u shares a fraction μ of its edges with vertices outside of C while the remaining $1 - \mu$ of its edges are shared with vertices within C. Thus, the mixing parameter μ controls the extent to which communities mix, with communities becoming less distinguishable as μ increases. Finally, in the LFR benchmark with overlap, the parameter $\rho \in (0, 1)$ is the proportion of vertices that are contained in exactly two communities, and therefore controls the extent of overlap. If u belongs to two communities in the overlapping LFR benchmark, then μ represents the proportion of edges of u that fall outside all these communities.

5.2. *Background benchmarks*. To assess detection methods on networks with background, we propose three principled test bed simulations: (1) a network with no communities (and therefore all vertices are background); (2) a network with a single embedded community; and (3) a network with disjoint communities and background. In what follows, we first describe how to simulate each type of network and then discuss the results for each type.

Networks with no community structure: It is important to measure the extent to which a detection method correctly identifies the lack of community structure when none is present. We construct such background networks by using two random network models: the Erdős–Rényi model of Erdős and Rényi (1960) where all vertices are linked with equal probability, and the configuration model of Molloy

and Reed (1995) where vertices are linked according to a prescribed degree sequence as discussed in Section 2.

For each of these models, we vary the size n and mean degree $\overline{D}$ in order to control the edge density of the generated network. In particular, for configuration random networks, we specify that the degree sequence follows a power law with degree τ_1 and average degree $\overline{D}$.

Single embedded community: We consider networks that contain a single embedded community and many background vertices. To construct such networks, we use a variant of the stochastic two block model of Snijders and Nowicki (1997), that has a simple generative procedure. First, vertices are placed randomly and independently in two blocks, C_1 and C_2, according to the probabilities π_1 and $\pi_2 = 1 - \pi_1$. An edge is included between a pair of distinct vertices $u \in C_i$ and $v \in C_j$ with probability $P_{i,j}$, independently from pair to pair.

To construct a network of size n with a single embedded community C_1 and background C_2, we generate a stochastic two block model using $\boldsymbol{\pi} = \{\pi, 1 - \pi\}$ with $\pi \in (0, 1)$ and $\mathbf{P} = \{P_{i,j} : 1 \leq i, j \leq 2\}$ given by

$$\mathbf{P} = \theta \begin{pmatrix} \kappa & 1 \\ 1 & 1 \end{pmatrix}.$$

Here $\kappa > 1$ controls the inner community edge probability, and $\theta < 1$ controls the average degree of the network. Modifying π controls for the size of the embedded community. The parameters θ and n can be modified to control the edge density of the network. By generating a network of fixed size and mean degree, one can assess the sensitivity of a detection method by running the method across a range of π. We note that Zhao, Levina and Zhu (2011) used a similar benchmark network to assess the performance of their own detection algorithm.

Disjoint communities and background: As a third benchmark test set, we simulate a network with background and degree heterogeneities. To do so, we propose combining the LFR benchmark described in Section 5.1 with the block structure described above. We construct this network in two steps using the same parameters as the LFR benchmark described in Table 7. First, we independently and randomly assign vertices to one of two blocks C_1 and C_2 according to probabilities $\boldsymbol{\pi} = \{\pi, 1 - \pi\}$. We place edges between vertices in block C_1 according to the disjoint LFR benchmark with parameters $\Theta = (\tau_1, \tau_2, n \cdot \pi, \mu, \overline{D} \cdot \pi, [s_1, s_2])$. The remaining vertices, corresponding to C_2, are connected to all vertices with equal probability $P_2 := \overline{D}(1 - \pi)$. Thus, our benchmark is constructed as a stochastic 2 block model described by $\boldsymbol{\pi}$ and

$$\mathbf{P} = \begin{pmatrix} \mathbf{P}_{\text{LFR}} & P_2 \\ P_2 & P_2 \end{pmatrix},$$

where $\mathbf{P}_{\text{LFR}}$ denotes the edge probabilities between vertices in C_1 derived from the LFR random network. The resulting network has average degree $\overline{D}$. On average, a fraction π of the vertices exhibit community structure following the LFR

disjoint benchmark, while the remaining vertices are connected to each other and to vertices in the first block in an Erdős–Rényi like fashion. This new benchmark is flexible and can be used to assess the performance of any community detection method for networks with background.

5.3. *Results*. *Networks with no community structure*: We generated both Erdős–Rényi and configuration model random graphs with 1000 vertices, with average degree $\overline{D}$ ranging from 10 to 100 in increments of 10. The degree sequence of the vertices in the configuration network follow a power-law distribution with degree $\tau_1 = 2$. For each value of $\overline{D}$, we generate 30 random graphs, with edge probabilities determined by the value of $\overline{D}$. In each of the simulations, ESSC assigned all nodes to background, as desired.

Single embedded community: We generated networks of size 2000, and set κ to 10, so that the edge probability within the single community is ten times that of the background. We selected values of θ to generate networks with average degree $\overline{D}$ of 30, 40 and 50. For each value of $\overline{D}$, we generated networks with embedded communities of size $\pi * 2000$ for π ranging from 0.01 to 0.3.

For each set of parameters, we generated 30 network realizations and gave these as input to ESSC, Spectral, ZLZ and OSLOM. We set Spectral to partition the network into two communities and set ZLZ to extract one community, thereby giving both of the methods an advantage over the other methods considered.

In order to measure the ability of each method to find the true single embedded community, we used the maximum Jaccard Match score of the detected communities. In detail, we measured the Jaccard score between each detected community and the true embedded community and reported the maximum of these values for each simulation. Results are shown in Figure 7.

From Figure 7, we see that ESSC is able to find, with Match ≈ 1, single embedded communities even when the community is as small as 4% of the total network. As the size of embedded community increases, the performance of each method improves, eventually reaching near optimal performance. In the case of small embedded communities ($\pi < 0.05$), ESSC and ZLZ perform similarly, with ESSC having a slight advantage. Finally, ESSC and all other methods improve as the average degree of the network increases. Across all simulations, we note that OSLOM did not find more than two nontrivial communities.

Disjoint communities and background: We simulated networks of size $n = 2000$ with $\pi = 1/2$, so that half of the vertices were background and the other half belonged to disjoint communities generated according to the LFR benchmark. Networks were generated with average degree $\overline{D} = 30$, 40 and 50, with community sizes in the range $[s_1, s_2] = [20, 100]$. Degree distributions were generated according to a power law with degree exponent $\tau_1 = 2$ and community size distributions were generated according to a power law with degree exponent $\tau_2 = 1$. For each value of $\overline{D}$, networks were generated with mixing parameter μ ranging between

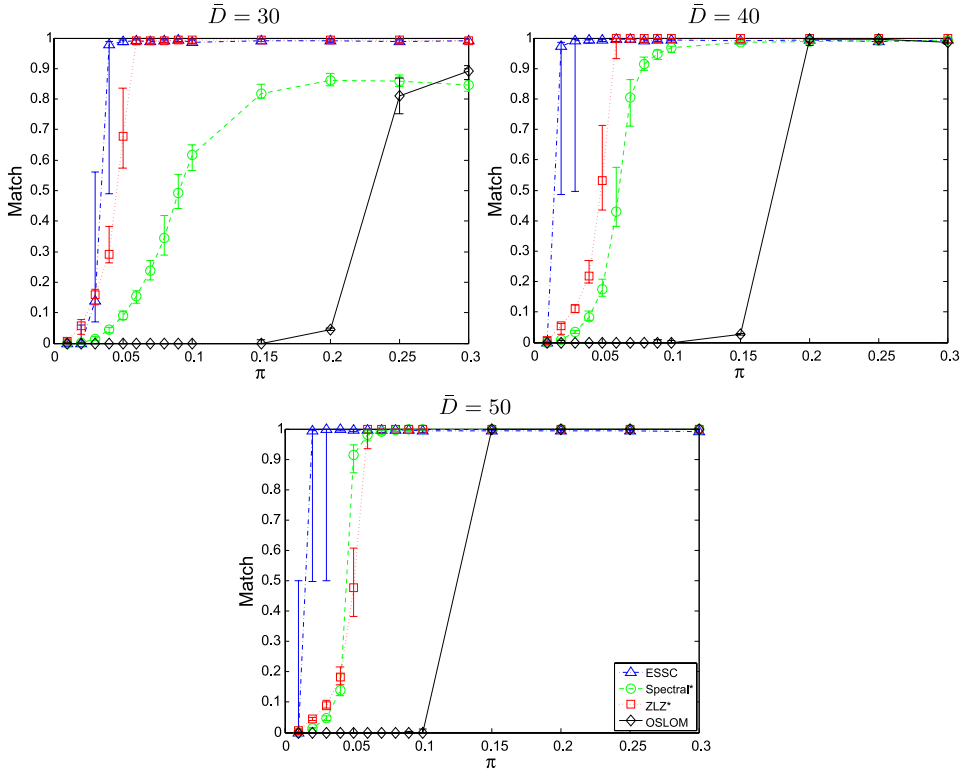

FIG. 7. *The results for networks with a single embedded community. Shown are the first, second and third quartile of the maximum Jaccard Match of each method over* 30 *realizations across values of* π. **Spectral and ZLZ were given the true number of communities*: *Spectral was set to partition the network into two communities, while ZLZ was set to extract* 1 *community.*

0.1 and 0.8 in increments of 0.1. For each set of parameters 30 network realizations were generated and then passed as input to ESSC, Spectral, ZLZ, OSLOM and Infomap. As before, the Spectral and ZLZ were run using the true number of communities. The generalized normalized mutual information (NMI) was used to measure the concordance of the detected communities and the true communities with background vertices treated as a single community. NMI is an information theoretic tool that can measure the similarity between two partitions as well as between two covers of a network. For more information on this similarity measure, refer to Lancichinetti, Fortunato and Kertész (2009). Results are shown in Figure 8.

Figure 8 tells us several interesting things about the performance of ESSC and other detection methods on complex networks with background. First, we see that ESSC performs well (NMI ≈ 1) across a range of mixing parameters μ from 0.1 to 0.5. After $\mu = 0.6$, ESSC finds no significant communities and, hence, the performance falls at this point. Infomap competes favorably with ESSC up until $\mu = 0.3$, at which point Infomap places all vertices in the same community. Interestingly,

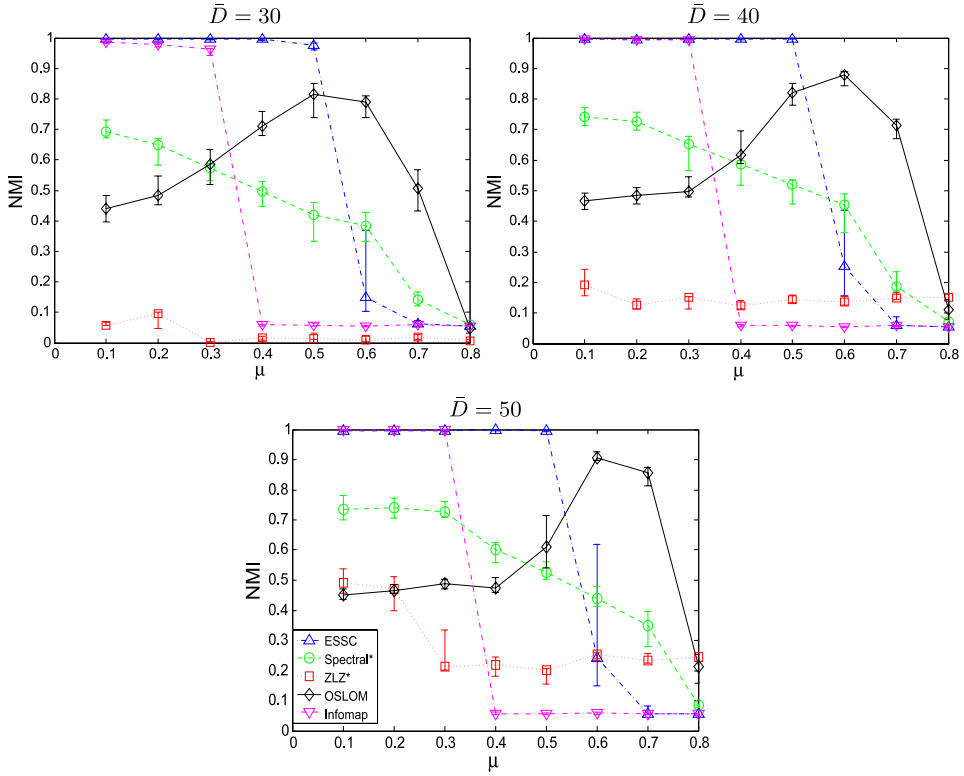

FIG. 8. *The results for networks with LFR and background features. Shown are the first, second and third quartile match of each method over* 30 *realizations across values of* μ. *The degree distribution of the significant community structure follows a power law with exponent* $\tau_1 = 2$ *with average degree* $\overline{D}$ *specified in each figure. *Here, Spectral and ZLZ were given the true number of communities.*

OSLOM has a peak of performance around $\mu = 0.6$. This appears to hinge on the fact that the method measures the strength of a community through assuming that vertices outside a community are close to the connectivity of the vertex of the community that has the lowest connectivity for the specified community. Highly mixed communities tend to favor this similarity, giving OSLOM an advantage in these cases. Importantly, ESSC performs nearly as well on networks of disjoint communities *with* background vertices as it does on these types of networks *without* background (see the Appendix B for nonbackground simulations). On the other hand, the remaining methods tend to, on average, perform much worse when background vertices are introduced.

6. Discussion. The identification of communities of tightly connected vertices in networks has proven to be an important tool in the exploratory analysis and study of a variety of complex connected systems. In this paper we introduced a means to measure the statistical significance of connection between a single vertex

and any collection of vertices in undirected networks through a reference distribution derived from the properties of the conditional configuration model. We introduced and evaluated a testing based community detection method, ESSC, which identifies statistically significant communities through the use of p-values derived from this reference distribution. This method automatically chooses the number of communities and relies only upon one parameter which guides the false discovery rate of discovered communities.

The ESSC extraction technique directly addresses the importance of identifying background vertices within a network that need not necessarily be assigned to identified communities. Given the heterogeneities of vertex roles in most real-world network data, identifying background nodes is an important aspect of community detection. Methods which identify background vertices can help prevent the noise associated with their connections from polluting the otherwise significant features among and between communities.

We evaluated ESSC and a number of competing community detection methods using a variety of quantitative and network-specific validation measures. We have shown that ESSC is able to capture features of network data that are relevant to the modeled complex system. For instance, in the Caltech network study we found that ESSC identified communities closely matching the dormitory residence of its individuals; similarly, in the political blog study ESSC identified communities matching the political affiliation of the bloggers in the network. Importantly, ESSC identified a moderate amount of background for each analyzed network in this paper, suggesting potential benefits to distinguishing background in a network.

Finally, through a series of simulations we have shown that ESSC is able to successfully capture both overlapping and disjoint community structure, as well as community structure in networks with background. In the former scenario, ESSC is competitive with many modern detection methods, while in the latter we find that ESSC outperforms competing methods.

The development of ESSC relied on undirected, unweighted networks, however, this can be extended to networks of different structures, including directed, multilayer and time-varying networks. Understanding the statistical significance of communities in each of these more complex network structures requires both theoretical and methodological work, providing avenues for future research. This includes comparing ESSC to the various stochastic block model fitting algorithms and other permutation-based statistical methods that have been recently developed over the past few years. Furthermore, understanding the consistency properties of the ESSC algorithm is an interesting question of independent interest which will require recently developed probabilistic tools.

APPENDIX A: APPROXIMATE DISTRIBUTION OF $\hat{d}(u : B)$

Here we state and prove Theorem 1 which gives the approximate law of $\hat{d}_n(u : B)$ on which our algorithm is based in the large network limit. The result is specific

to the conditional configuration model, which we use as a null network model in order to find significant community structure.

PROOF OF THEOREM 1. Equation (2.2) implies that for the number of edges $E_{o,n}$ one has

$$\int_{\mathbb{R}} x\, dF_n(x) = \sum_{k=0}^{\infty} k \frac{N_k(n)}{n} = 2\frac{|E_{o,n}|}{n} \sim \mu,$$

where $N_k(n)$ is the number of vertices of degree k. Thus, $|E_{o,n}| \sim n\mu/2$.

Now to understand the distribution of $\hat{d}_n(u : B)$, namely, the number of connections of vertex u to the subset B in $\mathrm{CM}(\mathbf{d}_{o,n})$, we use the fact that for constructing the configuration model, one can start at any vertex and start sequentially attaching the half-edges of that vertex at random to available half-edges. We start with the fixed vertex u and decide the half-edges paired to the $d_{o,n}(u) := k$ half-edges of vertex u. Write A_1 for the event that the first half-edge of vertex v connects to the set B and write $r_1(B)$ for the probability of this event. Then,

$$r_1(B) = \frac{\sum_{v\in B} d_{o,n}(v)}{[\sum_{v\in[n]} d_{o,n}(v)] - 1} = \frac{\sum_{v\in B} d_{o,n}(v)}{2|E_{o,n}| - 1}. \tag{A.1}$$

Now if each half-stub sampled with replacement from the stubs corresponding to set B, then $\hat{d}_n(u : B)$ would exactly correspond to a Binomial distribution. The main issue to understand is the effect of sampling without replacement from the half-stubs of B, namely, once a half-stub of B is used by u, it cannot be reused. In general, for $1 \le i \le k$, let A_i denote the event that half-edge i connects to the set B and write $r_i(B)$ for the conditional probability of A_i conditional on the outcomes of the first $i-1$ choices. For $i = 2$, we claim that uniformly on all outcomes for the first edge, this conditional probability can be bounded as

$$\frac{[\sum_{v\in B} d_{o,n}(v)] - 1}{2|E_{o,n}| - 2} \le r_2(B) \le \frac{\sum_{v\in B} d_{o,n}(v)}{2|E_{o,n}| - 2}. \tag{A.2}$$

The lower bound arises if the first half-edge of v connected to a half-edge of B, while the upper bound arises if the first half-edge does not connect to a half-edge emanating from B. Arguing analogously for $1 \le i \le k$, we find that the conditional probability $r_i(B)$ that the ith half-edge of vertex v connects to B is bounded (uniformly on all choices of the first $i-1$ edges) as

$$\frac{[\sum_{v\in B} d_{o,n}(v)] - (i-1)}{2|E_{o,n}| - i} \le r_i(B) \le \frac{\sum_{v\in B} d_{o,n}(v)}{2|E_{o,n}| - i}. \tag{A.3}$$

Recall that $p_n(B) = \sum_{v\in B} d_{o,n}(v)/2|E_{o,n}|$. Since $|E_{o,n}| \sim n\mu/2$, using (A.3), we have

$$\sup_{1\le i\le k} |r_i(B) - p_n(B)| \le 3\frac{k}{2|E_{o,n}|} + O\left(\left(\frac{k}{2|E_{o,n}|}\right)^2\right) \to 0 \tag{A.4}$$

as $n \to \infty$.

Now note that the random variable of interest $\hat{d}_n(u : B)$ can be expressed as

$$\hat{d}_n(u : B) = \sum_{i=1}^{k} \mathbb{1}\{A_i\}.$$

Equation (A.4) implies that

$$d_{\mathrm{TV}}\big(\hat{d}_n(u : B), \mathrm{Bin}\big(k, p_n(B)\big)\big) \to 0 \qquad \text{as } n \to \infty.$$

This completes the proof. □

APPENDIX B: SIMULATIONS ON DISJOINT AND OVERLAPPING COMMUNITY BENCHMARKS

Disjoint communities: LFR benchmarks of size 2000 were simulated with two ranges of community size, [10, 50] (small, S) and [20, 100] (big, B), where the community sizes were derived from a power-law distribution with exponent $\tau_2 = 1$ and with average degree $\overline{D}$ equal to 30, 40 or 50 with degrees deriving from a power-law distribution with exponent $\tau_1 = 2$. For each value of $\overline{D}$, networks were generated with values of μ ranging from 0.1 to 0.8 in increments of 0.1. Thirty realizations were generated from each set of parameters, and the resulting networks were input to the ESSC, GenLouvain, Infomap, OSLOM and Spectral methods. For Spectral, the parameter k was set to the true number of communities, thereby providing Spectral with an advantage over the other methods considered. Normalized mutual information (NMI) [Lancichinetti and Fortunato (2009b)] was used as a measure of performance for all methods. The results are summarized in Figure 9.

ESSC performs well (NMI ≈ 1) for all simulations with mixing parameter $\mu \leq 0.6$. In networks with small communities ([10, 50]), ESSC finds no significant communities for extreme values of μ (≥ 0.7). In networks with larger communities ([20, 100]), ESSC identifies underlying structure when $\mu = 0.7$, and performs particularly well for dense networks ($\overline{D} = 40, 50$). These results suggests that, when communities are weakly defined, ESSC performs better when the underlying communities are large. Overall, ESSC, OSLOM and Infomap performed ideally when $\mu \leq 0.6$.

Overlapping communities: LFR benchmarks of size 2000 were simulated with two ranges of community size, [10, 50] (small, S) and [20, 100] (big, B), with size distribution following a power law with exponent $\tau_2 = 1$ and with average degree $\overline{D}$ equal to 30, 40 or 50 where the degree distribution follows a power law with exponent $\tau_1 = 2$. For each value of $\overline{D}$, networks were generated with values of ρ ranging from 0.1 to 0.8 in increments of 0.1. The mixing parameter μ was set to 0.3. Thirty realizations were generated from each set of parameters and then input

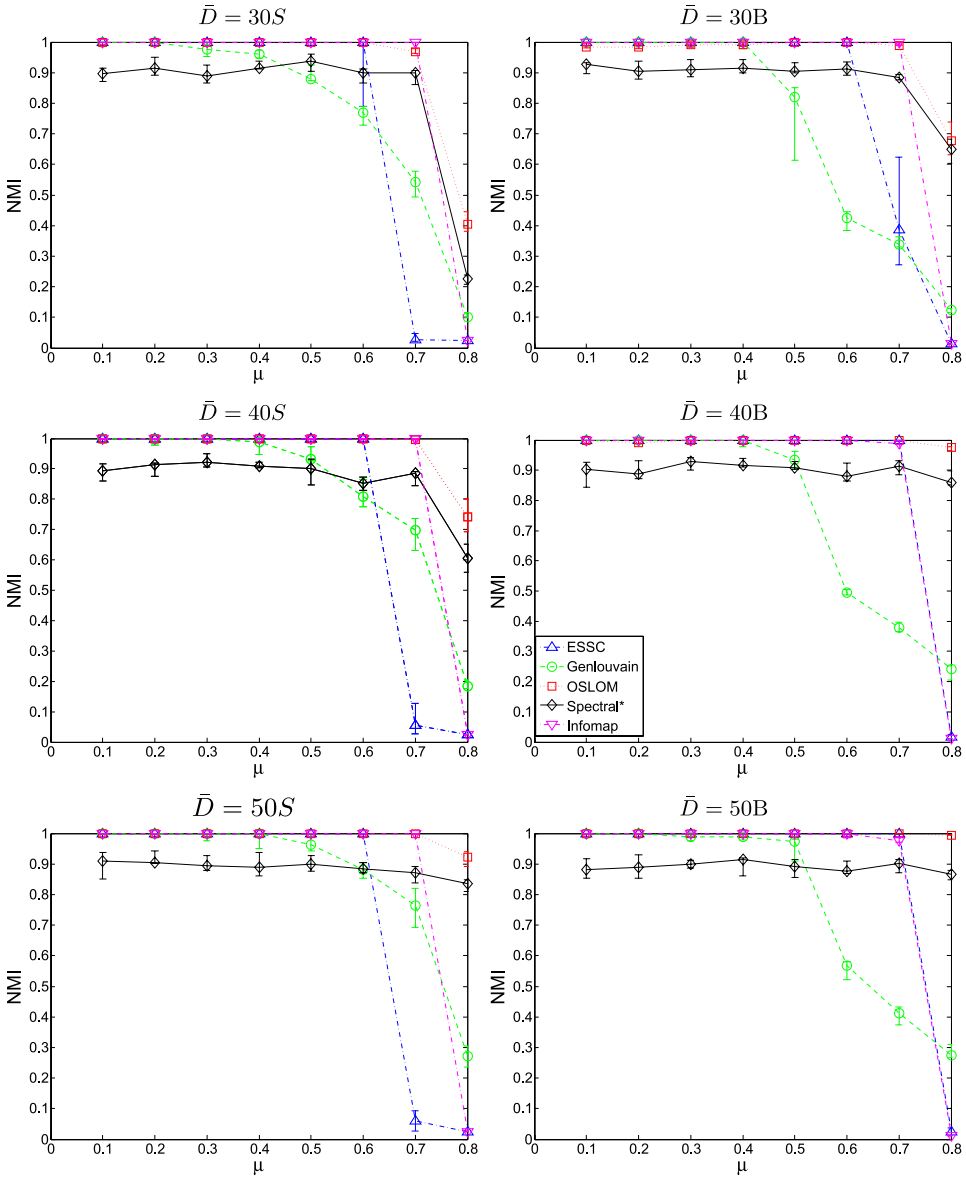

FIG. 9. *The results on the LFR disjoint benchmarks. Shown are the first, second and third quartile match of each method over* 30 *realizations across values of* μ. *The degree distribution follows a power law with exponent* $\tau_1 = 2$ *with average degree specified in each plot. *Here, Spectral was given the true number of communities.*

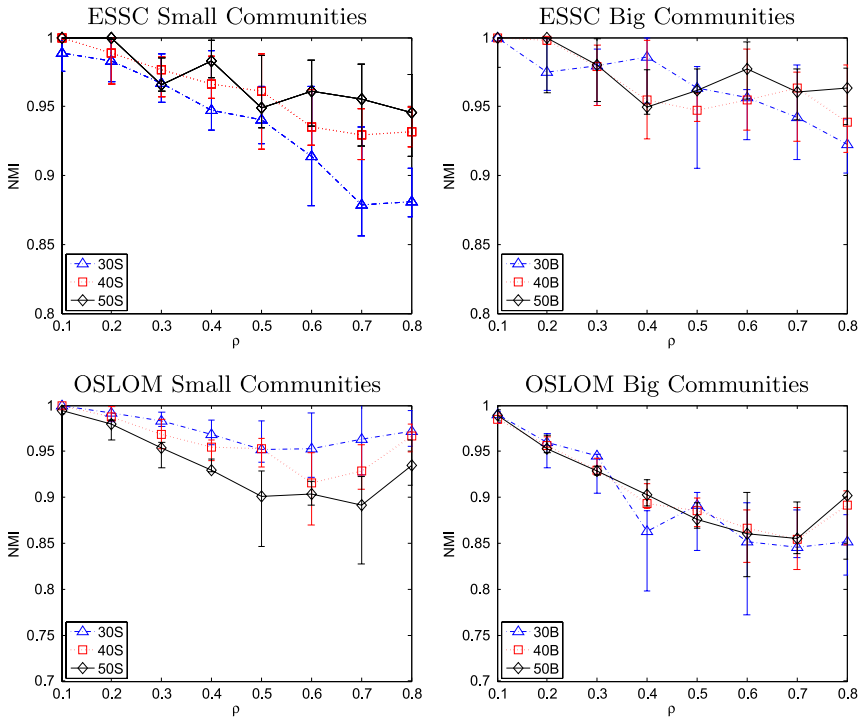

FIG. 10. *The results on the LFR overlapping benchmarks. Shown are the first, second and third quartile match of each method over* 30 *realizations across values of* ρ *at fixed* $\mu = 0.3$ *for both small* [30–50] *and big* [50–100] *communities. The degree distribution follows a power law with exponent* $\tau_1 = 2$ *with average degree specified by the color of each line.*

to ESSC and OSLOM. Once again, the generalized NMI was used to evaluate the similarity between the detected communities and the true cover. The results are summarized in Figure 10.

From Figure 10, we first notice that ESSC performs competitively with OSLOM in detecting overlapping community structure across all ρ. In networks with small communities (size in [10, 50]), the performance of ESSC improves as the density of the network increases. We also see that ESSC improves when the size of the communities increases as observed by comparing the left and right panels of the ESSC results in Figure 10. This agrees with our observation in the disjoint community study suggesting that ESSC prefers networks with larger communities.

APPENDIX C: PARAMETER SETTINGS OF DETECTION METHODS

We now describe the exact parameter settings as well as the code used for all detection methods throughout our real network analysis and simulation studies in Sections 4–5:

- *ESSC*: We use the MATLAB implementation of the algorithm provided by the authors at http://www.unc.edu/~jameswd/research.html. We set α to be 0.05 for all real data sets and simulated networks except for the Caltech Facebook network where we set α to be 0.01.
- *OSLOM*: We use the C++ implementation available at http://www.oslom.org/software.htm. For each study we use the default settings under an unweighted undirected network with no hierarchy. The p-value threshold is by default set at 0.1. A random seed is used for its random number generator.
- *Infomap*: We use the C++ implementation available at http://www.mapequation.org/code.html. For each study we use the default settings of the algorithm for an undirected network. We use a random positive integer as the seed and run 500 attempts of the algorithm to partition the network.
- *k-means*: We use the MATLAB implementation of the algorithm that is available for current MATLAB software. In each study we choose k according to the network as described throughout the text. We ran the algorithm over 500 iterations and used a random seed for initialization.
- *Spectral clustering*: We use the MATLAB implementation of the normalized Spectral Clustering algorithm. We choose k according to the network as described in the text. Again, we ran the algorithm over 500 iterations and used a random seed for initialization.
- *GenLouvain*: We use the MATLAB implementation of the generalized version of Louvain (GenLouvain) from Jutla, Jeub and Mucha (2011/2012). For the real network analysis, we run the algorithm across a range of resolution parameters, γ ranging from 0.1 to 1.0 (in increments of 0.1). For each γ, we look at the number of communities of the resulting partition and choose γ to be the first value for which the size is stable in terms of being constant across neighboring values of γ. In doing so, we chose $\gamma = 0.8$ for the Caltech Facebook network and $\gamma = 0.3$ for the political blog network. In the simulation study, we use the randomized version of GenLouvain (available on the same website) and choose the partition of the highest modularity across 30 repetitions. In each run, we use the default resolution parameter $\gamma = 1$. We use a random seed for each run of the algorithm.
- *ZLZ*: We use the R implementation provided to us by the author Yunpeng Zhao. We run the tabu search part of the algorithm 1000 iterations for each run. We choose k according to the network as described in our report. The normalized default score from Zhao, Levina and Zhu (2011) was used as the objective function to which the algorithm was run to optimize. A random seed was set for initialization.

APPENDIX D: ON THE EFFECTS OF α

As discussed in the main paper, α is the only tunable parameter of the ESSC algorithm. The value of α controls the level for which communities are declared statistically signficant. To get an idea of how sensitive the algorithm is to this

TABLE 8
A summary of the communities detected by ESSC across a range of values of α when run on the Caltech Facebook network. These statistics are the same as those presented in Section 4

α	N_C	$\overline{S}$	$\hat{\sigma}_S$	$\overline{\mathbb{M}}$	$\overline{D}_{\text{in}}$	$\overline{D}_{\text{out}}$	P_B
0.01	7	78.57	16.03	1.03	55.76	15.81	0.30
0.02	7	80.29	15.52	1.04	55.52	14.97	0.29
0.03	7	82.43	15.05	1.05	55.14	14.41	0.28
0.04	6	86.67	12.40	1.02	56.34	17.98	0.33
0.05	6	94.33	14.02	1.07	55.25	17.33	0.30
0.06	6	95.67	14.12	1.07	54.92	17.26	0.30
0.07	6	97.33	14.99	1.07	54.58	16.04	0.28
0.08	6	98.17	14.93	1.07	54.16	16.93	0.28
0.09	8	110.63	22.61	1.28	52.38	7.42	0.19
0.10	8	117.13	31.02	1.36	51.95	9.50	0.19

parameter, we run the algorithm on the first two analyzed data sets—the Caltech Facebook network and the political blog network—with values of α between 0.01 and 0.10. We summarize the detected communities using the statistics of Section 4. A summary of results are provided in Tables 8 and 10. The match of the identified communities with those discussed in the main text are given in Tables 9 and 11. Further, we assess the similarity of the background vertices from each setting using the Jaccard score. The match and statistics are shown below. In general, these statistics suggest that the communities detected by the ESSC algorithm are robust in the sense that they are not sensitive to the choice of α.

TABLE 9
The Jaccard score of the background vertices distinguished at each value of α when compared to the background vertices found with $\alpha = 0.01$. These analyses are done on the Caltech Facebook presented in Section 4

α	**Jaccard score**
0.01	1.00
0.02	0.9652
0.03	0.9304
0.04	0.8015
0.05	0.7303
0.06	0.7116
0.07	0.6985
0.08	0.7011
0.09	0.5907
0.10	0.6085

TABLE 10
A summary of the communities detected by ESSC across a range of values of α when run on the political blog network. These statistics are the same as those presented in Section 4

α	N_C	$\overline{S}$	$\hat{\sigma}_S$	$\overline{\mathbb{M}}$	$\overline{D}_{\text{in}}$	$\overline{D}_{\text{out}}$	P_B
0.01	2	394.5	54.45	1.00	40.51	3.40	0.35
0.02	2	406.5	67.18	1.00	39.47	3.27	0.33
0.03	2	420.0	53.74	1.00	38.40	3.07	0.31
0.04	2	423.5	57.28	1.00	38.14	3.00	0.31
0.05	2	448.5	75.66	1.00	36.30	2.58	0.27
0.06	2	449.5	75.66	1.00	36.27	2.45	0.26
0.07	2	431.0	46.67	1.00	37.60	2.84	0.29
0.08	3	528.3	146.92	1.30	27.37	24	0.01
0.09	2	449.5	72.83	1.00	36.24	2.54	0.26
0.10	3	323.67	249.93	1.02	34.39	2.56	0.22

Acknowledgments. We would like to thank the referees, Associate Editor and the Editor for their constructive suggestions which led to a significant improvement of the paper. We would like to thank Mason Porter for sharing the Caltech Facebook data set that we analyzed in Section 4.1. We would also like to thank Yunpeng Zhao for contributing his code for the ZLZ extraction algorithm.

TABLE 11
The Jaccard score of the background vertices distinguished at each value of α when compared to the background vertices found with $\alpha = 0.05$. These analyses are done on the political blog network of Section 4

α	**Jaccard score**
0.01	0.7483
0.02	0.7922
0.03	0.8433
0.04	0.8590
0.05	1.00
0.06	0.9938
0.07	0.8843
0.08	0.0062
0.09	0.9877
0.10	0.8277

SUPPLEMENTARY MATERIAL

Supplemental personal Facebook data set (DOI: 10.1214/14-AOAS760SUPP; .zip). We provide the personal Facebook data set as well as anonymized labels used in the analysis in Section 4.3 of the manuscript.

REFERENCES

ADAMIC, L. A. and GLANCE, N. (2005). The political blogosphere and the 2004 US election: Divided they blog. In *Proceedings of the* 3*rd International Workshop on Link Discovery* 36–43. ACM, New York.

AIROLDI, E. M., COSTA, T. B. and CHAN, S. H. (2013). Stochastic blockmodel approximation of a graphon: Theory and consistent estimation. In *Advances in Neural Information Processing Systems* 692–700.

AIROLDI, E. M., BLEI, D. M., FIENBERG, S. E. and XING, E. P. (2008). Mixed membership stochastic blockmodels. *J. Mach. Learn. Res.* **9** 1981–2014.

AMINI, A. A., CHEN, A., BICKEL, P. J. and LEVINA, E. (2013). Pseudo-likelihood methods for community detection in large sparse networks. *Ann. Statist.* **41** 2097–2122. MR3127859

BALL, B., KARRER, B. and NEWMAN, M. E. J. (2011). Efficient and principled method for detecting communities in networks. *Phys. Rev. E* (3) **84** 036103.

BARABÁSI, A.-L. and ALBERT, R. (1999). Emergence of scaling in random networks. *Science* **286** 509–512. MR2091634

BASSETT, D. S., WYMBS, N. F., PORTER, M. A., MUCHA, P. J., CARLSON, J. M. and GRAFTON, S. T. (2011). Dynamic reconfiguration of human brain networks during learning. *Proc. Natl. Acad. Sci. USA* **108** 7641–7646.

BENDER, E. A. and CANFIELD, E. R. (1978). The asymptotic number of labeled graphs with given degree sequences. *J. Combin. Theory Ser. A* **24** 296–307. MR0505796

BENJAMINI, Y. and HOCHBERG, Y. (1995). Controlling the false discovery rate: A practical and powerful approach to multiple testing. *J. R. Stat. Soc. Ser. B Stat. Methodol.* **57** 289–300. MR1325392

BICKEL, P. J. and CHEN, A. (2009). A nonparametric view of network models and Newman–Girvan and other modularities. *Proc. Natl. Acad. Sci. USA* **106** 21068–21073.

BLONDEL, VI. D., GUILLAUME, J. L., LAMBIOTTE, R. and LEFEBVRE, E. (2008). Fast unfolding of communities in large networks. *J. Stat. Mech. Theory Exp.* **2008** P10008.

BOLLOBÁS, B. (1979). A probabilistic proof of an asymptotic formula for the number of labelled regular graphs. Aarhus Universitet.

CLAUSET, A., MOORE, C. and NEWMAN, M. E. (2008). Hierarchical structure and the prediction of missing links in networks. *Nature* **453** 98–101.

CLAUSET, A., NEWMAN, M. E. J. and MOORE, C. (2004). Finding community structure in very large networks. *Phys. Rev. E* (3) **70** 066111.

DECELLE, A., KRZAKALA, F., MOORE, C. and ZDEBOROVÁ, L. (2011). Inference and phase transitions in the detection of modules in sparse networks. *Phys. Rev. Lett.* **107** 065701.

ERDŐS, P. and RÉNYI, A. (1960). On the evolution of random graphs. *Magyar Tud. Akad. Mat. Kutató Int. Közl.* **5** 17–61. MR0125031

ESTER, M., KRIEGEL, H.-P., SANDER, J. and XU, X. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. In *KDD* **96** 226–231.

FORTUNATO, S. (2010). Community detection in graphs. *Phys. Rep.* **486** 75–174. MR2580414

FREUND, Y. and SCHAPIRE, R. E. (1997). A decision-theoretic generalization of on-line learning and an application to boosting. *J. Comput. System Sci.* **55** 119–139. MR1473055

GIRVAN, M. and NEWMAN, M. E. J. (2002). Community structure in social and biological networks. *Proc. Natl. Acad. Sci. USA* **99** 7821–7826 (electronic). MR1908073

GLOVER, F. (1989). Tabu search—part I. *ORSA Journal on Computing* **1** 190–206.

GOLDBERG, A. V. and TARJAN, R. E. (1988). A new approach to the maximum-flow problem. *J. Assoc. Comput. Mach.* **35** 921–940. MR1072405

GOLDENBERG, A., ZHENG, A. X., FIENBERG, S. E. and AIROLDI, E. M. (2010). A survey of statistical network models. *Foundations and Trends in Machine Learning* **2** 129–233.

GREENE, D., DOYLE, D. and CUNNINGHAM, P. (2010). Tracking the evolution of communities in dynamic social networks. In *International Conference on Advances in Social Networks Analysis and Mining* (*ASONAM*) 176–183. Springer, New York.

HANDCOCK, M. S., RAFTERY, A. E. and TANTRUM, J. M. (2007). Model-based clustering for social networks. *J. Roy. Statist. Soc. Ser. A* **170** 301–354. MR2364300

HASTIE, T., TIBSHIRANI, R. and FRIEDMAN, J. (2001). *The Elements of Statistical Learning*: *Data Mining*, *Inference*, *and Prediction*. Springer, New York. MR1851606

HINNEBURG, A. and KEIM, D. A. (1998). An efficient approach to clustering in large multimedia databases with noise. In *KDD*, 1998 58–65.

HOFF, P. D., RAFTERY, A. E. and HANDCOCK, M. S. (2002). Latent space approaches to social network analysis. *J. Amer. Statist. Assoc.* **97** 1090–1098. MR1951262

HOLLAND, P. W., LASKEY, K. B. and LEINHARDT, S. (1983). Stochastic blockmodels: First steps. *Social Networks* **5** 109–137. MR0718088

JUTLA, I. S., JEUB, L. G. S. and MUCHA, P. J. (2011/2012). A generalized Louvain method for community detection implemented in MATLAB. Available at http://netwiki.amath.unc.edu/GenLouvain.

KRZAKALA, F., MOORE, C., MOSSEL, E., NEEMAN, J., SLY, A., ZDEBOROVÁ, L. and ZHANG, P. (2013). Spectral redemption: Clustering sparse networks. Preprint. Available at arXiv:1306.5550.

LANCICHINETTI, A. and FORTUNATO, S. (2009a). Benchmarks for testing community detection algorithms on directed and weighted graphs with overlapping communities. *Phys. Rev. E* (3) **80** 016118.

LANCICHINETTI, A. and FORTUNATO, S. (2009b). Community detection algorithms: A comparative analysis. *Phys. Rev. E* (3) **80** 056117.

LANCICHINETTI, A., FORTUNATO, S. and KERTÉSZ, J. (2009). Detecting the overlapping and hierarchical community structure in complex networks. *New J. Phys.* **11** 033015.

LANCICHINETTI, A., RADICCHI, F., RAMASCO, J. J. and FORTUNATO, S. (2011). Finding statistically significant communities in networks. *PloS One* **6** e18961.

LEE, C. and CUNNINGHAM, P. (2013). Benchmarking community detection methods on social media data. Preprint. Available at arXiv:1302.0739.

LESKOVEC, J., LANG, K. J., DASGUPTA, A. and MAHONEY, M. W. (2009). Community structure in large networks: Natural cluster sizes and the absence of large well-defined clusters. *Internet Math.* **6** 29–123. MR2736090

LEWIS, A. C., JONES, N. S., PORTER, M. A. and DEANE, C. M. (2010). The function of communities in protein interaction networks at multiple scales. *BMC Systems Biology* **4** 1–14.

MÉZARD, M. and MONTANARI, A. (2009). *Information*, *Physics*, *and Computation*. Oxford Univ. Press, Oxford. MR2518205

MIRITELLO, G., MORO, E. and LARA, R. (2011). Dynamical strength of social ties in information spreading. *Phys. Rev. E* (3) **83** 045102.

MOLLOY, M. and REED, B. (1995). A critical point for random graphs with a given degree sequence. *Random Structures Algorithms* **6** 161–179.

MUCHA, P. J., RICHARDSON, T., MACON, K., PORTER, M. A. and ONNELA, J.-P. (2010). Community structure in time-dependent, multiscale, and multiplex networks. *Science* **328** 876–878. MR2662590

MUHAMMAD, S. A. and VAN LAERHOVEN, K. (2013). Quantitative analysis of community detection methods for longitudinal mobile data. In *International Conference on Social Intelligence and Technology* (*SOCIETY*) 47–56. Springer, New York.

NEWMAN, M. E. J. (2006). Modularity and community structure in networks. *Proc. Natl. Acad. Sci. USA* **103** 8577–8582.

NEWMAN, M. E. J. and GIRVAN, M. (2004). Finding and evaluating community structure in networks. *Phys. Rev. E* (3) **69** 026113.

NG, A. Y., JORDAN, M. I. and WEISS, Y. (2002). On spectral clustering: Analysis and an algorithm. *Adv. Neural Inf. Process. Syst.* **2** 849–856.

NOWICKI, K. and SNIJDERS, T. A. B. (2001). Estimation and prediction for stochastic blockstructures. *J. Amer. Statist. Assoc.* **96** 1077–1087. MR1947255

OLHEDE, S. C. and WOLFE, P. J. (2013). Network histograms and universality of blockmodel approximation. Preprint. Available at arXiv:1312.5306.

ONNELA, J.-P., ARBESMAN, S., GONZÁLEZ, M. C., BARABÁSI, A.-L. and CHRISTAKIS, N. A. (2011). Geographic constraints on social network groups. *PLoS ONE* **6** e16939.

PAPADOPOULOS, S., KOMPATSIARIS, Y., VAKALI, A. and SPYRIDONOS, P. (2012). Community detection in social media. *Data Min. Knowl. Discov.* **24** 515–554.

PORTER, M. A., ONNELA, J.-P. and MUCHA, P. J. (2009). Communities in networks. *Notices Amer. Math. Soc.* **56** 1082–1097. MR2568495

ROSVALL, M., AXELSSON, D. and BERGSTROM, C. T. (2009). The map equation. *The European Physical Journal Special Topics* **178** 13–23.

ROSVALL, M. and BERGSTROM, C. T. (2008). Maps of random walks on complex networks reveal community structure. *Proc. Natl. Acad. Sci. USA* **105** 1118–1123.

ROSVALL, M. and BERGSTROM, C. T. (2010). Mapping change in large networks. *PLoS ONE* **5** e8694.

SHABALIN, A. A., WEIGMAN, V. J., PEROU, C. M. and NOBEL, A. B. (2009). Finding large average submatrices in high dimensional data. *Ann. Appl. Stat.* **3** 985–1012. MR2750383

SHI, J. and MALIK, J. (2000). Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **22** 888–905.

SNIJDERS, T. A. B. and NOWICKI, K. (1997). Estimation and prediction for stochastic blockmodels for graphs with latent block structure. *J. Classification* **14** 75–100. MR1449742

TRAUD, A. L., MUCHA, P. J. and PORTER, M. A. (2012). Social structure of Facebook networks. *Phys. A*: *Statistical Mechanics and Its Applications* **391** 4165–4180.

TRAUD, A. L., KELSIC, E. D., MUCHA, P. J. and PORTER, M. A. (2011). Comparing community structure to characteristics in online collegiate social networks. *SIAM Rev.* **53** 526–543. MR2834086

WEI, Y. C. and CHENG, C. K. (1989). Towards efficient hierarchical designs by ratio cut partitioning. In *IEEE International Conference on Computer-Aided Design* (*ICCAD*-89). *Digest of Technical papers* 298–301. IEEE, New York.

WILSON, J., WANG, S., MUCHA, P., BHAMIDI, S. and NOBEL, A. (2014). Supplement to "A testing based extraction algorithm for identifying significant communities in networks." DOI:10.1214/14-AOAS760SUPP.

XIE, J., KELLEY, S. and SZYMANSKI, B. K. (2011). Overlapping community detection in networks: The state of the art and comparative study. Preprint. Available at arXiv:1110.5813.

YANG, J. and LESKOVEC, J. (2012). Defining and Evaluating Network Communities based on Ground-truth. In *Proceedings of the ACM SIGKDD Workshop on Data Semantics*, 2012. ACM, New York.

ZHAO, Y., LEVINA, E. and ZHU, J. (2011). Community extraction for social networks. *Proc. Natl. Acad. Sci. USA* **108** 7321–7326.

ZHAO, Y., LEVINA, E. and ZHU, J. (2012). Consistency of community detection in networks under degree-corrected stochastic block models. *Ann. Statist.* **40** 2266–2292. MR3059083

J. D. WILSON
S. BHAMIDI
A. B. NOBEL
DEPARTMENT OF STATISTICS
AND OPERATIONS RESEARCH
UNIVERSITY OF NORTH CAROLINA AT CHAPEL HILL
CHAPEL HILL, NORTH CAROLINA 27599
USA
E-MAIL: jameswd@email.unc.edu
bhamidi@email.unc.edu
nobel@email.unc.edu

P. J. MUCHA
DEPARTMENT OF APPLIED PHYSICAL SCIENCES
UNIVERSITY OF NORTH CAROLINA
AT CHAPEL HILL
CHAPEL HILL, NORTH CAROLINA 25799
USA
E-MAIL: mucha@email.unc.edu

S. WANG
DEPARTMENT OF MATHEMATICS
UNIVERSITY OF NORTH CAROLINA AT CHAPEL HILL
CHAPEL HILL, NORTH CAROLINA 25799
USA
E-MAIL: wangsimi@email.unc.edu