Supplement to: Significance-based community detection in weighted networks

John Palowitch*  
Shankar Bhamidi  
Andrew B. Nobel  
Department of Statistics and Operations Research  
University of North Carolina at Chapel Hill  
Chapel Hill, NC 27599

Editor:

1. Simulation Framework Preliminaries

In this section and the following sections we describe the benchmarking simulation framework used for the performance analysis of CCME and other methods in Section 6 of the main document. In Table 1, we list and name the complete list of parameters controlling the simulated networks:

Table 1: Simulation model parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$n$</td>
<td>Number of nodes in communities</td>
</tr>
<tr>
<td>$n_b$</td>
<td>Number of nodes in background</td>
</tr>
<tr>
<td>$m_{\text{max}}$</td>
<td>Max community size</td>
</tr>
<tr>
<td>$m_{\text{min}}$</td>
<td>Min community size</td>
</tr>
<tr>
<td>$\tau_1$</td>
<td>Power-law for degree parameters</td>
</tr>
<tr>
<td>$\tau_2$</td>
<td>Power-law for community sizes</td>
</tr>
<tr>
<td>$k$</td>
<td>Mean of degree parameter power-law</td>
</tr>
<tr>
<td>$k_{\text{max}}$</td>
<td>Maximum degree parameter</td>
</tr>
<tr>
<td>$s_c$</td>
<td>Within-community edge signal</td>
</tr>
<tr>
<td>$s_w$</td>
<td>Within-community weight signal</td>
</tr>
<tr>
<td>$o_n$</td>
<td>Number of nodes in multiple communities</td>
</tr>
<tr>
<td>$o_m$</td>
<td>Number of memberships for overlap nodes</td>
</tr>
<tr>
<td>$F$</td>
<td>Distributions of edge weights</td>
</tr>
<tr>
<td>$\sigma^2$</td>
<td>Variance parameter for $P$</td>
</tr>
<tr>
<td>$\beta$</td>
<td>Power-law for strength parameters</td>
</tr>
</tbody>
</table>

2. Simulation of community nodes

The framework is capable of simulating networks with or without background nodes. For now, we describe the simulation procedure without background nodes, i.e. with $n_b = 0$. Later, we describe how to simulate a network with background nodes, which involves a slight modification to the procedure in this section. Regardless of the presence of background nodes, the first step is to determine community sizes and node memberships.

2.1 Community structure and node degree/strength parameters

In this section we describe how to obtain the community assignments of the $n$ community-nodes. The goal is to obtain a cover $\mathcal{C} := \{C_1, \ldots, C_K\}$ of the nodes $N$. The following steps to obtain $\mathcal{C}$ are almost exactly as those from the benchmark in Lancichinetti and Fortunato (2009), used extensively in Lancichinetti et al. (2011) and Xie et al. (2013).
1. Each of the \( o_n \) overlapping node will have \( o_m \) memberships. Let \( n_m := n + o_n(o_m - 1) \) be the number of node memberships present in the network.

2. Draw community sizes from a power law with maximum value \( m_{\text{max}} \), minimum value \( m_{\text{min}} \), and exponent \( -\tau_2 \), until the sum of community sizes is greater than or equal to \( n_m \). If the sum is greater than \( n_m \), we reduce the sizes of the communities proportionally until the sum is equal to \( n_m \).

3. Form a bipartite graph of community markers on one side and node markers on the other. Each community marker has number of empty node slots given by step (b), and each node has a number of memberships given by step (a). Sequentially pair node memberships and community node slots uniformly at random, without replacement, until every node membership is paired with a community. This process is a bipartite version of the standard configuration model. For more details, see Lancichinetti and Fortunato (2009).

With the community assignments in hand, simulation of the network proceeds according to the Weighted Stochastic Block Model as outlined in Section 6 from the main text. We describe choices for particular components of this model in the following section.

### 2.2 Simulation of edges and weights

As described in Section 6, we set the \( P \) and \( M \) matrices to have diagonals equal to \( s_e \) and \( s_w \) (respectively, see Table 1), and off-diagonals equal to 1. We note that this homogeneity facilitates creating networks with overlapping communities. With variance in the diagonal of \( P \), for example, it would not be obvious with what probability to connect overlapping nodes that overlap to two of the same communities, simultaneously. It remains to obtain the strength and degree propensity parameters \( \psi \) and \( \phi \); we do so analogously to the simulation framework in Lancichinetti et al. (2011). We first draw \( \phi \) from a power law with exponent \( \tau_1 \), mean \( k \), and maximum \( k_{\text{max}} \) (see Table 1). Next we set \( \psi \) by the formula \( \psi(u) = \phi(u)^{\beta+1} \) (this is mentioned in Section 6).

It is worth noting here that, under the model given below, the expected degree of node \( u \) is approximately \( \phi(u) \) and the expected strength approximately \( \psi(u) \). Therefore, heterogeneity/skewness in \( \phi \) and \( \psi \) induce heterogeneity/skewness in the degrees and strengths of the simulated networks. However, for reasons that will be made clear in the sections to follow, we prefer to have (at least) the total expected degree and total expected strength of the simulated networks match \( \phi_T \) and \( \psi_T \), respectively. As such, after drawing \( \phi \) from its power law and determining \( \psi \) from the aforementioned formula, we scale these vectors so their sums match the total expected degree and strength of \( G \). The scaling constants depend on \( P \) and \( M \) and are easily derivable from the model’s generative algorithm (described in Section 4.2.1 of the main text).

### 2.3 Parameter settings

Here we list the “default” settings of the simulation model, mentioned in Section 6 of the main text. The following choices for parameters were made regardless of the simulation setting: \( \tau_2 = -2 \), \( k = \sqrt{n} \), \( k_{\text{max}} = 3k \) (three settings which make the degree/strength
distributions skewed and the network sparse), $\beta = 0.5$ (to induce a non-trivial power law between strengths and degrees), $\tau_1 = -1$, $m_{\text{min}} = n/5$, $m_{\text{max}} = 3m_{\text{max}}/2$ (settings which produce between 3 and 7 communities per network with skewed size distribution), and $\sigma^2 = 1/2$. Other parameter choices are specific to the simulation settings, and described in Section 6 of the main text.

3. Background node simulation

If $n_b > 0$, we generate a network with $n$ community nodes, and then add $n_b$ background nodes, generating all remaining edges and weights according to the continuous configuration null model introduced in the main text. First, we obtain node-wise parameters for all $n + n_b$ nodes, yielding vectors $\phi$ and $\psi$ as in Section 2. In a simulated network without background, $\phi(u)$ and $\psi(u)$ are approximately $E[d(u)]$ and $E[s(u)]$, respectively. To ensure that this remains the case in a network for which background nodes are added after the simulation of community nodes, we must split up each $\phi(u)$ and $\psi(u)$ into community and background portions. A few other adjustments must also be made after the simulation of community nodes. To this end, define

- $N_C := \{1, \ldots, n\}; \ N_B := \{n + 1, \ldots, n + n_b\} \rightarrow$ community and background node sets
- $\phi_{C,T} := \sum_{N_C} \phi(u); \ \phi_{B,T} := \sum_{N_B} \phi(u) \rightarrow$ target total degrees of community and background nodes
- $\phi_C(u) := \frac{\phi_{C,T}}{\phi_{T}} \phi(u); \ \phi_B(u) := \frac{\phi_{B,T}}{\phi_{T}} \phi(u) \rightarrow$ target edge-counts between $u$ and the community and background nodes
- $\phi_{1,T} := \sum_{N_C} \phi_C(u); \ \phi_{2,T} := \sum_{N_B} \phi_B(u) \rightarrow$ target total degrees of community and background subnetworks
- $d_C^o(u) := \sum_{v \in N_C} A_{uv}; \ d_B^o(u) := \sum_{v \in N_B} A_{uv} \rightarrow$ observed edge-counts between $u$ and the community and background nodes

The above definitions exist analogously for the strength parameters $\psi$ (replacing “$d$” with “$s$” where appropriate). The word “target” indicates that we will set up the background simulation model so that these values are the approximate expected values of the graph statistics they represent.

3.1 Adjusted community-node simulation model

The only adjustment to be made to the simulation of community nodes, described in Section 2.2, is that the degree and strength parameters are set to a certain fraction of their original values. This accounts for the eventual addition of background nodes, where the remaining (random) part of each nodes degree and strength is to be simulated. So, the
community-node simulation (if background nodes are to be added later) follows the process described in Section 2 with degree parameters \( \{ \phi_C(1), \ldots, \phi_C(n) \} \) and strength parameters \( \{ \psi_C(1) \ldots \psi_C(n) \} \).

### 3.2 Edges and weights for background

For the simulation of the background nodes (following the community nodes) our goal is to specify adjusted degree/strength parameters \( \phi' \) and \( \psi' \) given the observed edge-sums \( \{ d^o_C(1), \ldots, d^o_C(n) \} \) and weight-sums \( \{ s^o_C(1), \ldots, s^o_C(n) \} \) from the community nodes. In what follows we describe this specification for \( \phi' \) only; the specification for \( \psi' \) is exactly analogous. We first represent \( \phi'_T \), which we have yet to determine, into community and background totals:

\[
\phi'_T = \phi'_{C,T} + \phi'_{B,T}
\]

Since the background subnetwork has not yet been generated, we make the specification \( \phi'(u) := \phi(u) \) for all \( u \in N_B \), and hence \( \phi'_{B,T} = \phi_{B,T} \) is known. To address \( \phi'_{C,T} \), note that for each community node \( u \in N_C \), \( \phi'(u) \) may be represented similarly:

\[
\phi'(u) = \frac{\phi_B(u)}{\phi_T} \phi(u), \quad u \in N_B
\]

This reduces the problem of specifying \( \phi'(u) \) to specifying \( \phi'_C(u) \) and \( \phi'_B(u) \). Since the community node subnetwork has already been generated, we set \( \phi'_C(u) := d^o_C(u) \). Next, recalling that \( \phi_B(u) := \frac{\phi_B(u)}{\phi_T} \phi(u) \), we make the specification \( \phi'_B(u) := \frac{\phi_B(u)}{\phi_T} \phi(u) \) (which must be solved for via \( \phi'_T \), in the following). So, in total, we have

\[
\phi'_T := \sum_{u \in N_C \cup N_B} \phi'(u)
\]

\[
= \sum_{u \in N_C} \left[ d^o_C(u) + \frac{\phi_B(u)}{\phi_T} \phi(u) \right] + \sum_{u \in N_B} \phi(u)
\]

\[
= d^o_{C,T} + \frac{\phi_B(u)}{\phi_T} \phi_C + \phi_{B,T}
\]

Where \( d^o_{C,T} := \sum_{u \in N_C} d^o_C(u) \). The solution for \( \phi'_T \) from this quadratic is

\[
\phi'_T = \frac{\phi_B(u) + d^o_{C,T}}{2} + \sqrt{\left(\frac{\phi_B(u) + d^o_{C,T}}{2} \right)^2 + \phi_C \phi_{B,T}}
\]

which then immediately gives the full vector \( \phi' \). We can now simulate the remaining edges in the network. Specifically, for each \( u \in N_B \) and each \( v \in N_C \cup N_B \), we simulate an edge according to

\[
P(A_{uv} = 1) = \frac{\phi'(u) \phi'(v)}{\phi'_T} \text{ independent across node pairs}
\]
We solve for $\psi'$ analogously. Then for each $u \in N_B$ and each $v \in N$, we simulate an edge weight according to

$$W_{uv} = \begin{cases} f_{uv}(\phi', \psi')\xi_{uv}, & A_{uv} = 1 \\ 0, & A_{uv} = 1 \end{cases}$$

where $\xi \sim F$, is as it was for the generation of the community node subnetwork.

The above simulation steps correspond precisely to the continuous configuration model with parameters $(\phi', \psi', P, \theta)$. Some basic computational trials have shown that, for large networks, the solution for $\phi_T'$ is quite close to $\phi_T$. Therefore, for each $u \in N_B$, $\mathbb{E}(d(u))$ is almost exactly $\phi(u)$, i.e. what it would be under the model in 2.2, without background nodes. The same holds for the strengths and expected strengths. Together with equation 2, this implies the background nodes are behaving according to the continuous configuration model, even as they are a sub-network within a larger network with communities.

To illustrate these points, we simulated a sample network from the default framework with parameters $n = 5,000$, $n_b = 1,000$, $s_e = s_w = 3$, disjoint communities, and other parameters specified by 2.3. These settings are akin to what was used in Section 6 of the main text. First we plotted $\phi'$ and $\psi'$ against the empirical strengths and degrees with lowess curves to check the match. Figure 1 shows the match is quite close. Second, for each node $u \in N$ and for each node block $B$ (either a true community or the background node set) we may calculate an empirical $z$-score for $s(u, B, G)$, as described in Section 4.1 of the main text. The $z$-score for $s(u, B, G)$ is a measure of connection significance, with respect to the continuous configuration model, between $u$ and $B$. Let $K$ be the number of true communities in the network. For each $i, j = 1, \ldots, K + 1$, where $K + 1$ is the index of the background node block, we computed the empirical average of $z$-statistics between nodes $u$ from node block $i$ the node block $B$ corresponding to index $j$. Theses empirical averages can be arranged in a $(K + 1) \times (K + 1)$ matrix showing the average inter-block connectivities of the network. In Figure 2 we display a visualization of this matrix, which shows preferential connection within communities, and roughly null connection between the background nodes and all blocks.

Figure 1: Empirical degrees/strengths vs. adjusted parameters for the example network
Figure 2: Average empirical z-statistics between nodes and node blocks

References


Figure 3: SLPAw, OSLOM, and CCME results from January and February 2015 U.S. airport networks. Maps created with ggmap (Kahle and Wickham, 2013)
Figure 4: SLPAw, OSLOM, and CCME results from March and April 2015 U.S. airport networks. Maps created with ggmap (Kahle and Wickham, 2013)
Figure 5: SLP\textsuperscript{Aw}, OSLOM, and CCME results from May and July U.S. airport networks. Maps created with \texttt{ggmap} (Kahle and Wickham, 2013)