

HHS Public Access

Author manuscript *J Complex Netw.* Author manuscript; available in PMC 2018 May 29.

Published in final edited form as:

J Complex Netw. 2017 December ; 5(6): 839–857. doi:10.1093/comnet/cnx014.

Generating Bipartite Networks with a Prescribed Joint Degree Distribution

Asma Azizi^{*},

Department of Mathematics, Tulane University, New Orleans, LA 70118, USA

Jeremy Dewar,

Department of Mathematics, Tulane University, New Orleans, LA 70118, USA

Tong Wu, and

Department of mathematics, North Carolina State University, Raleigh, NC 27695, USA

James M. Hyman

Department of Mathematics, Tulane University, New Orleans, LA 70118, USA

Abstract

We describe a class of new algorithms to construct bipartite networks that preserves a prescribed degree and joint-degree (degree-degree) distribution of the nodes. Bipartite networks are graphs that can represent real-world interactions between two disjoint sets, such as actor-movie networks, author-article networks, co-occurrence networks, and heterosexual partnership networks. Often there is a strong correlation between the degree of a node and the degrees of the neighbors of that node that must be preserved when generating a network that reflects the structure of the underling system. Our bipartite 2K(B2K) algorithms generate an ensemble of networks that preserve prescribed degree sequences for the two disjoint set of nodes in the bipartite network, and the joint-degree distribution that is the distribution of the algorithms on a romance network using the NetworkX software environment to compare other properties of a target network that are not directly enforced by the B2K algorithms. We observe that when average degree of nodes is low, as is the case for romance and heterosexual partnership networks, then the B2K networks tend to preserve additional properties, such as the cluster coefficients, than algorithms that do not preserve the joint-degree distribution of the original network.

Keywords

Bipartite random network; Joint-degree distribution; dk network generation algorithm; 2K algorithm; JDM matrix; BJD matrix

2000 Math Subject Classification

34K30; 35K57; 35Q80; 92D25

1. Introduction

A *bipartite network*, sometimes called *two-mode network* or *affiliation network*, is a network whose nodes can be divided into two disjoint sets, upper and lower nodes, where every edge connects an upper node to a lower node. There are no edges between upper nodes, and no edges between lower nodes. Bipartite networks can provide an insightful representation of the interactions between two disjoint groups, with applications ranging from heterosexual partnerships [? ?], citation/collaboration networks [7], and ecological networks [12].

To avoid artifacts caused by the structure of a single network, mathematical models on these networks must quantify the range of possible predictions based on an ensemble of networks with properties similar to those in real-world networks. This requires algorithms to generate an ensemble of random networks that capture the relevant structure of these real-world networks. For many models, these important properties include the network degree distribution and joint-degree distribution which is sometimes called the degree correlation or degree-degree distribution. That is, if a property of a network, such as the degree and joint-degree distribution, affects the solution of questions being asked, such as how fast a disease will spread among a heterosexual population, then that property must be preserved in the networks used in the simulations.

For example, in heterosexual partnership networks the number of partners a man, or a woman, has is correlated with the number of partners that their partner has. **That is, most people with few partners might select partners who also have few partners, and most people with many partners might select partners who also have many partners. However, there might exist people with many partners whose partners have few partners.** Sex workers are an example of this situation. These statistical properties can often be captured in the heterosexual network by preserving the joint-degree distribution when simple rule-based assumptions, such as preferential attachment, fail to account for these complex correlations.

We recognize that the degree and joint-degree distributions are just two of many possible important properties of a network that can affect the solution of a network-based model. In **particular, when modeling the spread of information on a network, it seems likely that the joint degree-cluster distribution can influence the local saturation of information.** In unipartite networks, there is often an **assortativity tendency** where nodes are connected to nodes with similar properties. In bipartite networks, this affect is more often captured by a one-hop associative mixing where the a node is more like the neighbors of their neighbors, than they are like their neighbors. That is, in a heterosexual network, a woman is more like the partners of her partner than she is like her partner, or in an author/article network, the authors are more like the other authors of their papers than they are like the article.

Typical network generation algorithms that preserve the degree distributions are based on *stochastic, rewiring*, or *reconfiguration* approaches. The Erd s-Rényi random graph generation algorithm [13] is an example of stochastic approach where every two nodes are connected with probability p defined by the average degree of nodes in the network divided by their size. This approach is easily generalized to match a given degree distribution [6] or

joint-degree distribution [4]. The pseudograph reconfiguration algorithm [22] reproduces the given degree distribution exactly, however it may end up with a multigraph, a network with self-loop or multiple edges between two nodes, therefore, rewiring is needed to convert the network to a simple one. The rewiring approach takes a connected network and selects two random edges to reconnect, while preserving the average degree or degree distribution. Fully generating a network via rewiring does converge, although there is little analysis on the convergence rate [21].

The joint-degree distribution is correlated with the structural and dynamical properties of networks [10, 15, 24, 32]. This information is quantified in the symmetric joint-degree matrix (JDM) whose (i, j) element is the number of edges between nodes of degree i and nodes of degree j [31]. The necessary and sufficient condition for a simple network to exist for a given JDM is given by the Erd s–Gallai type theorem [1, 8, 31]:

Theorem 1.1—(Erd s–Gallai Type Theorem for JDM) Consider a network where M is the largest degree of the nodes in the network, then there is a simple network that has an $M \times M$ symmetric JDM if and only if

- 1. $n_i = \frac{1}{i}(JDM(i, i) + \sum_{j=1}^{M} JDM(i, j))$ is the number of nodes with degree *i* for $i = 1 \cdots M$.
- 2. $JDM(i, i) \leq \binom{n_i}{2}$, for $i = 1 \cdots M$.
- **3.** $JDM(i, j) \quad n_i n_j$, for i = j.

There are several methods, called 2K network generation algorithms, for generating unipartite networks, networks with only one type of node, that preserve both a given degree and joint-degree distributions. [4, 10, 15, 20, 24, 32].

Mahavedan et al. [20] have extended the rewiring approach to generate random networks using joint-degree distribution. They use the term 2K-series to introduce joint-degree distribution, and they compare stochastic, pseudograph, matching and rewiring and the extended pseudograph algorithms to construct networks using 2K-series. They compare the topology of networks made based on different algorithms and suggest that 2K- series or joint-degree distribution is enough to reproduce most metrics of interest for the network. They then use a configuration model to generate a 2K-network with the prescribed JDM, however their network may end up with multiple edges between two nodes. Miller et al. and Newman et al. [??] use the configuration model approach in their 2K edge-based algorithms to generate a random unipartite network by randomly assigning edges to match the given joint-degree sequence. A balanced degree invariant algorithm is provided by [31] for constructing simple networks from a given JDM, and a Monte Carlo Markov Chain method is used for sampling the networks. Gjoka et al. [16] designed a new algorithm for constructing simple networks with a target JDM. Bassler et al. [2] described a JDM sampling algorithm for generating independent (uncorrelated) networks that match a specified joint-degree distribution. Their algorithm is exact, that is, it does not exclude classes of graphical realizations of a given JDM and can generate all possible 2K networks

in the limit of an infinite number of samples. None of the mentioned unipartite 2K algorithms distinguishes between different types of nodes, and therefore, cannot be directly applied to bipartite networks.

There are bipartite 1K(B1K) algorithms for generating bipartite network that preserve a prescribed degree distribution of the network [18, 23] and prevent edges between similar nodes. We extend these B1K methodologies to create bipartite 2K(B2K) algorithms that preserve both the degree and joint-degree distributions. Note that the bipartite joint degree (*BJD*) matrix of a bipartite network can be nonsymmetric and is not a square matrix if the maximum degree in two groups are not the same. We find and prove a similar necessary and sufficient condition as Erd s–Gallai Type Theorem on *BJD* for constructing simple bipartite network and then use *BJD* matrix as an input to construct network. We then describe new bipartite algorithms for generating these random networks and investigating how well they reproduce other properties, such as the bipartite clustering, observed in real-world networks. This family of algorithms are called *B2K* algorithms and they preserve a given degree and joint-degree distributions of the network.

2. Bipartite Network

A *bipartite network*, sometimes called *two-mode network* or *affiliation network*, is a network whose nodes can be divided into two disjoint sets \mathbf{V}^u and \mathbf{V}^l such that every edge connects a node in \mathbf{V}^u to one in \mathbf{V}^l , there is no edge between nodes in \mathbf{V}^u , and no edges between nodes in \mathbf{V}^l . This network is shown like $\mathbf{G} = (\mathbf{V}^u, \mathbf{V}^l, \mathbf{E})$ consisting of a set of $P^u = |\mathbf{V}^u|$ upper nodes, $\mathbf{V}^u = \{\mathbf{v}_i^u \mid i = 1, 2, 3, ..., P^u\}$, a set of $P^l = |\mathbf{V}^l|$ lower nodes, $\mathbf{V}^l = \{\mathbf{v}_i^l \mid i = 1, 2, 3, ..., P^l\}$, together with a binary adjacency relation defining the set of edges

 $\mathbf{E} = \{\mathbf{e}_{ij} = \mathbf{v}_i^u \mathbf{v}_j^l \mid i \in \{1, 2, 3, \dots P^u\}, j \in \{1, 2, 3, \dots P^l\}\}, \text{ where } \mathbf{e}_{ij} = \mathbf{v}_i^u \mathbf{v}_j^l \text{ denotes the edge}$ between node \mathbf{v}_i^u and node \mathbf{v}_i^l .

The *degree* of a node \mathbf{v}_i , deg(\mathbf{v}_i), is defined as the number of neighboring nodes connected to the node by an edge. The *degree distribution* d_k defines the number of nodes with degree *k*. The *joint-degree distribution* or sometimes called *degree-degree distribution* or *degree correlation* (*k*, *j*) is the number of nodes with degree *j* which are connected to nodes with degree *k*. A bipartite network **G** can be represented by the *Bipartite Joint Degree* or *BJD* matrix:

$$BJD_{G} = \begin{bmatrix} e_{11} & e_{12} & e_{13} & \cdots & e_{1l} \\ e_{21} & e_{22} & e_{23} & \cdots & e_{2l} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ e_{u1} & e_{u2} & e_{u3} & \cdots & e_{ul} \end{bmatrix},$$

where, *u* is the maximum degree in upper nodes, and *I* is the maximum degree in lower nodes, each element e_{ij} is the number of edges between upper nodes with degree *i* and lower nodes with degree *j*.

The degree distribution of network **G** is defined by the number of upper nodes, d_k^u , and lower nodes, d_k^l , with degree *k*:

$$d_k^u = \frac{\sum_{j=1}^l e_{kj}}{k}$$
, and $d_k^l = \frac{\sum_{i=1}^u e_{ik}}{k}$

The number of nodes with degree k is $d_k = d_k^u + d_k^l$.

A *BJD* matrix is consistent with a bipartite network if there exists at least one bipartite network with this joint degree distribution. For an example, consider the *BJD* matrix

$$BJD = \begin{bmatrix} 2 & 2 \\ 4 & 0 \end{bmatrix},$$

each entry in the matrix, (i, j), is an edge between an upper node with degree *i* and lower node with degree *j*. There are $\frac{2+2}{1} = 4$ upper nodes with degree 1, and $\frac{4+0}{2} = 2$ upper node with degree 2, $\frac{2+4}{1} = 6$ lower nodes with degree 1, and $\frac{2+0}{2} = 1$ lower node with degree 2. On the other hand, $e_{21} = 4$ means that four of the edges of the graph will connect an upper node of degree 2 to a lower node of degree 1, or $e_{22} = 0$ means there is no edge between upper and lower nodes with degree 2.

Theorem 2.1—Let *BJD* be a matrix,

$$BJD = \begin{cases} e_{11} \ e_{12} \ e_{13} \ \cdots \ e_{1l} \\ e_{21} \ e_{22} \ e_{23} \ \cdots \ e_{2l} \\ \vdots \ \vdots \ \vdots \ \ddots \ \vdots \\ e_{u1} \ e_{u2} \ e_{u3} \ \cdots \ e_{ul} \end{cases},$$

where $d_i^u = \frac{\sum_j e_{ij}}{i}$ is the number of upper nodes with degree *i*, and $d_j^l = \frac{\sum_i e_{ij}}{j}$ is the number of lower nodes with degree *j*. If we have $e_{ij} \le d_i^u d_j^l$ for i = 1, ..., u and j = 1, ..., l, then, there exist at least one simple network, a network without self-loops or multiple edges, as defined by the *BJD* matrix.

Proof: Suppose the *BJD* matrix satisfies the assumption of theorem, and the network **G** corresponding to *BJD* matrix has at most one edge between all nodes, except for two nodes, where there are two edges between the upper node **u** of degree *i* and the lower node **v** of degree *j*. There are d_i^u upper nodes with degree *i* and d_j^l lower nodes with degree *j*.

For the trivial case when $d_i^u = 1$ and $d_j^l = 1$, we have $e_{ij} = 1$ and there can not be a multiple edge between the sole node **u** with degree *i* and sole node **v** with degree *j*.

When either $d_i^u > 1$ or $d_j^l > 1$, then there is an upper node, \mathbf{u}' , with degree *i* and a lower node, \mathbf{v}' , with degree *j* that are not connected. This follows from the contradiction argument where if all nodes with degree *i* are connected to all nodes with degree *j* and \mathbf{u} is connected to \mathbf{v} with two edges, then $e_{ij} = d_i^u d_j^l + 1 \nleq d_i^u d_j^l$, which contradicts the assumption of theorem. Therefore, such \mathbf{u}' and \mathbf{v}' exist. These \mathbf{u}' may be the same as \mathbf{u} or \mathbf{v}' may be the same as \mathbf{v} , but both can not happen:

- Case 1: We consider the case that \mathbf{u}' is different from \mathbf{u} and \mathbf{v}' is different from \mathbf{v} . Because \mathbf{u} is connected to \mathbf{v} within two edges, therefore, it has *i*-1 disjoint neighbors, however, because all edges connected to \mathbf{u}' are simple, therefore, \mathbf{u}' has *i* disjoint neighbors, thus \mathbf{u}' has a neighbor, like \mathbf{w} , which is not a neighbor of \mathbf{u} . Also, for the same reason \mathbf{v} has *j*-1 disjoint neighbors and \mathbf{v}' has *j* disjoint neighbors, thus \mathbf{v}' has a neighbor, like \mathbf{w}' , which is not a neighbor of \mathbf{v} . When this happens, then we rewire the network by first removing one of the the double edges \mathbf{u} , as well as edges $\mathbf{u}'\mathbf{w}$ and $\mathbf{v}'\mathbf{w}'$, then adding the edges \mathbf{u} , $\mathbf{v}\mathbf{w}'$, and $\mathbf{u}'\mathbf{v}'$. Therefore, we have a simple network \mathbf{G} . The figure below illustrates this rewiring process.
- Case 2: If u' is the same as u, then v' has to be different from v, in that case, because v has j-1 disjoint neighbors and v' has j disjoint neighbors, then v' has a neighbor, w', which is not a neighbor of v. Therefore, we rewire the network by removing edges uv and w' v' and adding the edges uv' and w'v. The figure below illustrates this process. For the case when v' is the same as v we have similar approach.

2.1 Generating Bipartite Network

We introduce five B2K algorithms to construct simple bipartite networks for a given BJD matrix satisfying the assumptions of Theorem 2.1. These algorithms are categorized as either an *edge* or *node* algorithm, depending on the network generation process. Both approaches can be used to generate an ensemble of B2K networks. However, as we will show, the statistical properties of the networks for the different algorithms differ. That is, the different approaches have different biases in sampling the space of all feasible networks.

The algorithms begin by grouping nodes into upper and lower nodes and assigning each node a desired degree based on the *BJD* matrix. In the edge algorithms, we first choose an entry in the BJD matrix - a tuple of the desired degrees of the upper node and lower node. We find the list of pair of nodes that satisfies the conditions of the tuple. We choose one pair of nodes randomly from this list and attach the edge if one does not exist. If we cannot find a pair of nodes in the list that do not have an edge between them, we choose one at random and add a double-edge. We repeat adding edges until all edges are placed. If we attached a double-edge during the generation, we rewire the graph as described in subsection 2.2. The full algorithms for each approach to graph generation are provided in the Appendix.

- **Random Edge (RE):** Define the remaining *BJD* matrix *RBJD* matrix equal to *BJD* matrix. Choose one (i,j) randomly from the *RBJD*. We find a pair of upper and lower nodes with degrees i, j respectively and add an edge between them, then update $(i, j) \rightarrow (i, j)$ –1. This process continues for each edge until the *RBJD* becomes zero matrix.
- **Maximum Edge-Degree (ED_{max}):** Of the remaining edges choose an edge to add from those with $\max(d_{\max})$ where $d_{\max}((i, j)) := \max(i, j)$ at random until there are no remaining edges.
- **Total Edge-Degree (TED):** Of the remaining edges choose an edge to add from those with $\max(d_{\text{total}})$ where $d_{\text{total}}((i, j)) := i + j$ at random until there are no remaining edges. (*Note:* We observed no statistical differences between the TED and ED_{max} approaches and will only present results for the ED_{max} algorithm.)

The node algorithms start with the unconnected list of upper and lower nodes and iteratively add new edges based on current state of the nodes in the network:

- Maximum Node-Degree (ND_{max}): Choose from the nodes with the highest desired degree. Choose possible edges that the chosen node could have and select appropriate neighbors. Select all of the neighbors for the chosen node, then choose another node from those with the highest desired degree. Repeat until all edges are added.
- **Maximum Stub Minimum Node-Degree** $(S_{max}ND_{min})$: Find the nodes with the fewest placed edges the most stubs and sort them by their desired degree, and choose the one with the maximum desired degree. Choose possible edges that the chosen node could have and select appropriate neighbors and make the edge. Continue until all edges are added.

Because our experiments require that we calculate the degree of nodes attached to edge for each possible edge in the network, the full set of experiments run in $O(|E|P^{u}P^{l})$ time, where |E| is the number of edges, P^{u} number of upper nodes and P^{l} number of lower nodes in the network. Therefore, it is prohibitive when |E|, P^{u} , or P^{l} is large, however, for when the average degree $(\frac{2|E|}{P^{u}+P^{l}})$ of the network is not high then the methods are more feasible.

2.2 Rewiring Approach

During the construction of our network, it is possible that there are not valid nodes, **u** and **v**, each of valid desired degrees, *i* and *j* that do not already have an edge between them. Our options are to increase a node beyond its desired degree, or attach a multiple edge. Because the final network satisfies a certain BJD, we choose to simplify the generation by allowing multiple edges between nodes **u** and **v** as long as the network still satisfies (*i*, *j*). Once all of the edges are attached, we use rewiring to remove multiple edges, which maintains the proper edge count for each (*i*, *j*).

Our approach follows the proof of theorem 2.1. Suppose **G** is the bipartite network generated by one of the approaches defined in the last subsection, and suppose **G** has at least one multiple edge. We randomly start with one of the multiple edges, say edge attached to the nodes **u** and **v** and then we follow the rewiring process explained in the theorem 2.1. The detail of the algorithm can be found in the appendix, algorithm 6.

3. Numerical Examples

We evaluated the RE, ED_{max} , ND_{max} , and $S_{max}ND_{min}$ algorithms by generating an ensemble of random bipartite networks that preserve the *B2K* properties of small (Southern women) and medium size (Romance network, Malaria network) real-world bipartite networks. We then compared the ensemble of randomly generated networks using descriptive measures that were not imposed when generating the networks:

- *Nc*: the number of connected components of the network.
- *Sg*: the size of giant component (the biggest connected component) of the network.
- *Cl*: the bipartite clustering coefficient of the network [?]. The bipartite clustering coefficient for each node is the fraction of the number of quadruplets of nodes with four links in network to the number of quadruplets of nodes with at least three. The clustering coefficient for a network is the average of clustering coefficient for all nodes.
- *Rc*: the average redundancy coefficient of all nodes in the network. The redundancy coefficient of a node **u** is the ratio of the overlap of **u** to the maximum possible overlap of **u** according to its degree. The overlap of **u** is the number of pairs of neighbors that have mutual neighbors themselves, other than **u**. The redundancy coefficient for a network is the average of redundancy coefficients for all nodes [?].

3.1 Southern Women

The Southern women empirical network [11] in the right panel of Figure (3) reflects the attendance of 18 women in Natchez, MS, USA at 14 social events in the 1930*s* and was collected by ethnographers to examine the roles of race and class in dictating social interactions [19]. The *BJD* matrix of the Southern women network is

Table (1) compares the properties of the original network and the average of 10,000 random networks generated by each B2K algorithm. The redundancy coefficients of the networks generated by B2K algorithms and the original network are similar. However, the clustering coefficients for the generated networks differ by about 15% from the original network. The visual comparison of the original network and a typical B2K generated network in Figure 3 shows how similar the networks are.

Currently, there are no *B*2*K* NetworkX [?] algorithms that preserve the joint-degree distribution for bipartite networks. To investigate the differences in the graphs generated with *B*1*K* and *B*2*K* algorithms, we compared our *B*2*K* generating network algorithms with the Newman et al. configuration [23] and Havel-Hakimi [18] NetworkX *B*1*K* algorithms. Using each algorithm we generated 10,000 networks, and then we compare the average of properties in Table (2). Before showing the result, we mention that Havel-Hakimi algorithms create a unique graph [?] in which the highest degree node in the upper set tends to be connected to the highest degree node in lower set (Havel-Hakimi algorithm), or highest degree node in upper set tends to be connected to lowest degree node in upper set to alternatively the highest and the lowest degree nodes in lower set until all stubs are connected (Alternating Havel-Hakimi) [18]. Therefore, for this family all our generated networks have the same metrics.

The properties of generated networks for Southern women data in Table (2) show that the Nc and Sg for all algorithms is close to the original network. However, the Cl for all the generated networks is far from the Cl for the original network. The B1K algorithms do not preserve the Rc of the original network, while the Rc for the B2K algorithms is close to the original value, indicating a correlation between Rc and the joint-degree distribution.

3.2 Romance Network

The network of heterosexual contact depicted in Figure (4) describes the structure of the adolescent romantic and heterosexual network in a population of 573 students at Jefferson High [3]. The original network is not a bipartite network: there are two edges that link two men and two women, representing homosexual relationships. We remove these two edges so that we have a bipartite network.

The *BJD* matrix of the Romance network, BJD_R , has $k_w = 6$ rows, and $k_m = 8$ columns, where k_w is the maximum degree for women students and k_m is the maximum degree for men. Each element (i, j) is the number of edges between women with degree *i* and men with degree *j*. The *BJD_R* matrix determines the degree distribution for women and men: the number of women with degree *i* is summation of elements in *i*-th row divided by *i*, and the number of men with degree *j* is summation of elements in *j*-th column divided by *j*;

$$BJD_{R} = \begin{pmatrix} 63 & 56 & 30 & 16 & 2 & 0 & 0 & 2 \\ 46 & 40 & 25 & 14 & 1 & 0 & 0 & 4 \\ 23 & 20 & 18 & 8 & 1 & 0 & 0 & 2 \\ 26 & 24 & 21 & 4 & 1 & 0 & 0 & 0 \\ 8 & 9 & 1 & 2 & 0 & 0 & 0 & 0 \\ 4 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

This matrix shows many people with few partners, indicated by many nodes with a low degree (upper left corner of matrix) and only a few people with many partners, indicated by few high degree nodes. The average degree is very low. Using our algorithms we compare the properties of generated random networks and the original Romance network. We generate an ensemble of 10,000 networks using each algorithm, the algorithms succeeded in generating networks that preserved both the degree and joint-degree distributions in every simulation. We observed that the statistical properties of the networks for the B2Kalgorithms were different, as shown in Table (3). This Table lists the Nc, Sg, Cl, and Rc for the ensemble of generated networks. Note that the size of the giant component for the real network is noticeably above the mean size of giant components in the randomly generated networks, especially SmaxNDmin. The Cl and Rc of all the generated networks are similar to the original Romance network, although the RE networks are the closest. We also compare B2K generating network algorithms with B1K algorithms in NetworkX [?] by generating 10,000 networks using each algorithm and computing the average of properties for each of them. Table (4) lists some properties of real Romance network, B2K networks and B1K networks generated using Romance data.

The Figure (5) plots the distribution of properties of the 10,000 simulated networks from B2K algorithms. The clustering coefficient, subfigure (5a), is approximately normally distributed for all the B2K algorithms. We observed that the mean value for the $S_{max}ND_{min}$ generated networks is slightly smaller, indicating that joint-degree distribution may weakly correlated with the clustering coefficient. The distribution of redundancy coefficients for the networks is right skewed (subfigure 5b). The giant component and the number of connected components generated by the $S_{max}ND_{min}$ is smaller than it is for the other B2K algorithms and it is skewed to the right, while the other distributions are closer to symmetric.

The giant component of generated network using configuration model, Havel-Hakimi algorithm, RE algorithm, and real Romance network are shown in Figure (6). This Figure shows that the network generated by B2KRE algorithm is visually more similar to the real network more than B1K network generators in NetworkX. As expected, this example shows that the joint-degree distributions preserves more properties of the original network than the bipartite algorithms that just preserve the degree distribution.

3.3 Malaria Network

The last empirical network is the malaria parasite. The parasite evades the human immune system via a protein camouflage, which is encoded in var genes [19]. In order to create novel camouflages, var genes frequently recombine, which amounts to the constrained splicing and shuffling of genetic substrings, giving rise naturally to community structures [5]. Nodes

types correspond to genes and their constituent substrings, and each substring connects to every gene in which it is present [19].

The *BJD* matrix for this network is of size 24×35 , that all big elements are distributed on row 2 to 7 and all columns of the matrix, rest of the matrix are 0 or small numbers of order 1. The average degree for genes is 9.98 and that of substrings is 3.67. Using *BJD* matrix and *B2K* algorithms, we generate 3,000 random networks and compare their properties with the original network. Table (5) shows the result.

The original Malaria network and randomly generated networks have the same BJD matrix, but very different properties, Table (5). All of our B2K algorithms tended to generate networks that reproduced the Nc and Sg of the original network, Table 6. However, neither the B2K algorithms nor B1K algorithms captured Cl and Rc, indicting that these quantities might only be loosely correlated with the joint-degree distribution.

These results indicate that the Malaria network may not be a typical network with these degree and joint-degree distributions. If so, then additional properties of the network might be needed for an ensemble of randomly generated networks to exhibit similar properties of the original network.

The distribution of clustering coefficient of these 3,000 generated networks using different algorithms is shown in Figure (7). For $S_{max}ND_{min}$ and RE algorithms, we observe a unimodal distribution with a very low variance, however, ND_{max} and ED_{max} are near to normal with very different parameters, suggesting that joint-degree distribution is not enough to capture clustering coefficient for this Malaria data.

4. Summary and Discussion

Some properties of real-world systems such as degree distribution, joint-degree distribution, or the degree-cluster distributions affect the dynamics on the network. When these dynamics are being modeled, then these properties need to be preserved in an ensemble of random networks to prevent introducing artifacts into the simulations. We have partially addressed this problem by describing new B2K algorithms to construct bipartite networks that exactly match both given degree and joint-degree distributions. An ensemble of these B2K networks can then be used to more accurately study dynamical systems like the spread of infections or information, and to quantify the uncertainty in these predictions.

We analyzed the properties of an ensemble of networks with the same BJD generated by the B2K algorithms that shared the same joint-degree distribution as several real-world bipartite networks. We observed a tight distribution in the number of connected components and size of giant component for all the networks with the same BJD. The clustering and redundancy coefficients for networks with the same BJD had a narrow distribution for low-degree networks and broad distributions for high-degree networks. That is, in a typical bipartite heterosexual network with low average degree, the networks all had similar clustering coefficients. When the average degree of network was high, such as the Malaria network, the networks generated by the B2K algorithms had a wider range of clustering and redundancy

coefficients, indicating that more information than joint-degree distribution is needed to retain these properties.

Although our algorithm preserves the degree and joint-degree distributions of a bipartite system, there are other properties that can be important in preserving when generating an ensemble of bipartite networks. The joint degree-cluster distribution can influence the local spread of information or a disease in a network with highly connected neighborhoods. The joint-degree distribution in a heterosexual network can capture the effects where people are like their partners, as observed in homosexual networks. However, there are often third-order degree relationships in bipartite networks, such as a one-hop associative mixing properties, where a node is more like the neighbors of their neighbors, than they are like their neighbors. This one-hop assortivity is a weaker condition than generating a bipartite 3K network that exactly preserves these third-order distributions [20]. We are investigating 'soft constraint' algorithms to capture this preferential attachment by biasing the random selection process in our node-based algorithms. **One of our goals is to generalize this approach to account for associative mixing in social networks where nodes also have associated properties, such as age, race, income, or spatial location [?].**

Acknowledgments

The authors thank Aric Hagberg and an anonymous reviewer for their useful comments and suggestions. This work was supported by the endowment for the Evelyn and John G. Phillips Distinguished Chair in Mathematics at Tulane University and grants from the National Institutes of Health National Institute of Child Health and Human Development (R01HD086794) and the National Institute of General Medical Sciences program for Models of Infectious Disease Agent Study (U01GM097658). The content is solely the responsibility of the authors and does not necessarily represent the official views of the National Institutes of Health.

References

- Amanatidis G, Green B, Mihail M. Graphic realizations of joint-degree matrices. 2015 arXiv preprint arXiv:1509.07076.
- 2. Bassler KE, Del Genio CI, Erd s PL, Miklós I, Toroczkai Z. Exact sampling of graphs with prescribed degree correlations. New Journal of Physics. 2015; 17(8):083052.
- 3. Bearman PS, Moody J, Stovel K. Chains of affection: The structure of adolescent romantic and sexual networks1. American journal of sociology. 2004; 110(1):44–91.
- 4. Boguná M, Pastor-Satorras R. Class of correlated random networks with hidden variables. Physical Review E. 2003; 68(3):036112.
- Bull PC, Buckee CO, Kyes S, Kortok MM, Thathy V, Guyah B, Stoute JA, Newbold CI, Marsh K. Plasmodium falciparum antigenic variation. Mapping mosaic var gene sequences onto a network of shared, highly polymorphic sequence blocks. Molecular microbiology. 2008; 68(6):1519–1534. [PubMed: 18433451]
- 6. Chung F, Lu L. Connected components in random graphs with given expected degree sequences. Annals of combinatorics. 2002; 6(2):125–145.
- 7. Cimini G, Gabrielli A, Labini FS. The scientific competitiveness of nations. PloS one. 2014; 9(12):e113470. [PubMed: 25493626]
- Czabarka É, Dutle A, Erd s PL, Miklós I. On realizations of a joint degree matrix. Discrete Applied Mathematics. 2015a; 181:283–288.
- Czabarka E, Rauh J, Sadeghi K, Short T, Szekely LA. On the Number of Non-zero Elements of Joint Degree Vectors. 2015b arXiv preprint arXiv:1511.01035.
- D'Agostino G, Scala A, Zlati V, Caldarelli G. Robustness and assortativity for diffusion-like processes in scale-free networks. EPL (Europhysics Letters). 2012; 97(6):68006.

- 11. Davis, A., Gardner, BB., Gardner, MR. Deep South: A social anthropological study of caste and class. Univ of South Carolina Press; 2009.
- 12. Dormann CF, Fründ J, Blüthgen N, Gruber B. Indices, graphs and null models: analyzing bipartite ecological networks. 2009
- 13. Erd s P, Rényi A. On random graphs. Publicationes Mathematicae Debrecen. 1959; 6:290-297.
- Erds PL, Miklós I, Toroczkai Z. A decomposition based proof for fast mixing of a Markov chain over balanced realizations of a joint degree matrix. SIAM Journal on Discrete Mathematics. 2015; 29(1):481–499.
- Eubank S, Guclu H, Kumar VA, Marathe MV, Srinivasan A, Toroczkai Z, Wang N. Modelling disease outbreaks in realistic urban social networks. Nature. 2004; 429(6988):180–184. [PubMed: 15141212]
- Gjoka, M., Tillman, B., Markopoulou, A. Construction of simple graphs with a target joint degree matrix and beyond. Computer Communications (INFOCOM), 2015 IEEE Conference on; IEEE; 2015. p. 1553-1561.
- 17. Gjoka, M., Tillman, B., Markopoulou, A., Pagh, R. Efficient Construction of 2K+ Graphs. 2010.
- Hakimi SL. On realizability of a set of integers as degrees of the vertices of a linear graph. I. Journal of the Society for Industrial and Applied Mathematics. 1962; 10(3):496–506.
- 19. Larremore DB, Clauset A, Jacobs AZ. Efficiently inferring community structure in bipartite networks. Physical Review E. 2014; 90(1):012805.
- Mahadevan, P., Krioukov, D., Fall, K., Vahdat, A. ACM SIGCOMM Computer Communication Review. Vol. 36. ACM; 2006. Systematic topology analysis and generation using degree correlations; p. 135-146.
- Mihail, CGM., Zegura, E. The markov chain simulation method for generating connected power law random graphs. Proceedings of the FifthWorkshop on Algorithm Engineering and Experiments; SIAM; 2003. p. 16
- 22. Molloy M, Reed B. A critical point for random graphs with a given degree sequence. Random structures & algorithms. 1995; 6(2–3):161–180.
- 23. Newman ME. The structure and function of networks. Computer Physics Communications. 2002; 147(1):40–45.
- 24. Newman ME. Mixing patterns in networks. Physical Review E. 2003; 67(2):026126.
- 25. Newman ME, Strogatz SH, Watts DJ. Random graphs with arbitrary degree distributions and their applications. Physical review E. 2001; 64(2):026118.
- 26. Rao AR, Jana R, Bandyopadhyay S. A Markov chain Monte Carlo method for generating random (0, 1)-matrices with given marginals. Sankhy : The Indian Journal of Statistics, Series A. 1996:225–242.
- 27. Ray J, Pinar A, Seshadhri C. A stopping criterion for Markov chains when generating independent random graphs. Journal of Complex Networks. 2014:cnu041.
- 28. Saracco F, Di Clemente R, Gabrielli A, Squartini T. Randomizing bipartite networks: the case of the World Trade Web. Scientific reports. 2015:5.
- 29. Seufert M, Lange S, Hossfeld T. More than topology: Joint topology and attribute sampling and generation of social network graphs. Computer Communications. 2016; 73:176–187.
- Stanton, I., Pinar, A. Sampling graphs with a prescribed joint degree distribution using Markov chains. Proceedings of the Meeting on Algorithm Engineering & Experiments; Society for Industrial and Applied Mathematics; 2011. p. 127-138.
- Stanton I, Pinar A. Constructing and sampling graphs with a prescribed joint degree distribution. Journal of Experimental Algorithmics (JEA). 2012; 17:3–5.
- 32. Williams O, Del Genio CI. Degree correlations in directed scale-free networks. PloS one. 2014; 9(10):e110121. [PubMed: 25310101]
- 33. Wu, SF. Online Social Networks. 2016.

Appendix

This appendix describes algorithms for the generation of random bipartite network. The input for all generation network algorithms 1–5 is *BJD* matrix, each element (i, j) of this matrix is the number of edges between upper nodes with degree *i* and lower nodes with degree *j*. At the beginning of each algorithm every node is given a degree, $deg(\mathbf{u})$, and the potential edges of a node as its *stubs*, *stub*(\mathbf{u}):= $deg(\mathbf{u})$.

The first and simplest algorithm, Random Edge (**RE**), selects edges randomly and makes them between two random nodes.

Algorithm 1

Random Edge (RE)

while $BJD > 0$ do
Randomly select an element (<i>i</i> , <i>j</i>)
Randomly select an upper node u with degree <i>i</i> and <i>stub</i> (u) > 0.
Randomly select a lower node v with degree j and $stub(\mathbf{v}) > 0$.
Make edge \mathbf{uv} , $stub(\mathbf{u}) \leftarrow stub(\mathbf{u}) - 1$, and $stub(\mathbf{v}) \leftarrow stub(\mathbf{v}) - 1$.
$(i, j) \leftarrow (i, j) - 1.$
end while

The Maximum Edge Degree (ED+) selects edges based on degree of nodes attached to them.

Algorithm 2

Maximum Edge Degree (ED+)

1:	while $BJD > 0$ do
2:	For remaining $(i, j) > 0$, find <i>m</i> where m : = $\max_{\{i, j\} > 0} \{i, j\}$.
3:	Randomly select an element $(i, j) > 0$ where $i = m$ or $j = m$.
4:	Randomly select an upper node u with degree <i>i</i> and <i>stub</i> (u) > 0.
5:	Randomly select a lower node v with degree j and $stub(\mathbf{v}) > 0$.
6:	Make edge \mathbf{uv} , $stub(\mathbf{u}) \leftarrow stub(\mathbf{u}) - 1$, and $stub(\mathbf{v}) \leftarrow stub(\mathbf{v}) - 1$.
7:	$(i, j) \leftarrow (i, j) - 1.$
8:	end while

Total Edge Degree (**TED**), first makes an edge for nodes whose summation of their degree is highest among others.

Algorithm 3

Total Edge Degree (TED)

^{1:} while BJD > 0 do

2:	For remaining $(i, j) > 0$, find <i>m</i> where $m := \max_{\substack{(i, j)}} (i + j)$.
3:	Randomly select an element $(i, j) > 0$ where $i + j = m$.
4:	Randomly select an upper node \mathbf{u} with degree <i>i</i> and <i>stub</i> (\mathbf{u}) > 0.
5:	Randomly select a lower node \mathbf{v} with degree <i>j</i> and <i>stub</i> (\mathbf{v}) > 0.
6:	Make edge \mathbf{uv} , $stub(\mathbf{u}) \leftarrow stub(\mathbf{u}) - 1$, and $stub(\mathbf{v}) \leftarrow stub(\mathbf{v}) - 1$.
7:	$(i, j) \leftarrow (i, j) - 1.$
8:	end while

The algorithm **ND**+ finds the neighbors for the node with maximum degree, and then go to the next highest degree node.

Algorithm 4

Maximum Node Degree (ND+)

1:	while $BJD > 0$ do
2:	From remaining nodes with positive sub, find the one with highest degree: $\mathbf{u} = \max_{d \in g(\mathbf{v})} \{\mathbf{v}\}$
3:	while $stub(\mathbf{u}) > 0$ do
4:	if u is upper node with degree <i>i</i>
5:	then:
6:	From the row i of <i>BJD</i> matrix randomly select an element $(i, j) > 0$.
7:	Randomly select a lower node v with desired degree j and $stub(\mathbf{v}) > 0$.
8:	Make edge \mathbf{uv} , $stub(\mathbf{u}) \leftarrow stub(\mathbf{u}) - 1$, and $stub(\mathbf{v}) \leftarrow stub(\mathbf{v}) - 1$, and $(i, j) \leftarrow (i, j) - 1$.
9:	
10:	if \mathbf{u} is lower node with degree j
11:	then:
12:	From the column j of <i>BJD</i> matrix randomly select an element $(i, j) > 0$.
13:	Randomly select an upper node v with desired degree <i>i</i> and $stub(\mathbf{v}) > 0$.
14:	Make edge \mathbf{uv} , $stub(\mathbf{u}) \leftarrow stub(\mathbf{u}) - 1$, and $stub(\mathbf{v}) \leftarrow stub(\mathbf{v}) - 1$, and $(i, j) \leftarrow (i, j) - 1$.
15:	
16:	end if
17:	end while
18:	end while

The algorithm $S_{max}ND_{min}$ captures social behavior more than the others, it always start with the highest degree node and end up with the highest degree node.

Algorithm 5

Maximum Stub Minimum Degree (SmaxNDmin)

2:

From remaining nodes, find the node \mathbf{u} with $\mathbf{u} = \min_{deg(\mathbf{v})} \{\max_{stub(\mathbf{v})} \{\mathbf{v}\}\}$.

^{1:} while BJD > 0 do

3:	if u is upper node with degree <i>i</i>
4:	then:
5:	From the row i of <i>BJD</i> matrix randomly select an element $(i, j) > 0$.
6:	Randomly select a lower node v with desired degree j and $stub(\mathbf{v}) > 0$.
7:	Make edge \mathbf{uv} , $stub(\mathbf{u}) \leftarrow stub(\mathbf{u}) - 1$, and $stub(\mathbf{v}) \leftarrow stub(\mathbf{v}) - 1$, and $(i, j) \leftarrow (i, j) - 1$.
8:	
9:	if \mathbf{u} is lower node with degree j
10:	then:
11:	From the column j of <i>BJD</i> matrix randomly select an element $(i, j) > 0$.
12:	Randomly select an upper node v with desired degree <i>i</i> and $stub(\mathbf{v}) > 0$.
13:	Make edge \mathbf{uv} , $stub(\mathbf{u}) \leftarrow stub(\mathbf{u}) - 1$, and $stub(\mathbf{v}) \leftarrow stub(\mathbf{v}) - 1$, and $(i, j) \leftarrow (i, j) - 1$.
14:	
15:	end if
16:	end while

The algorithms 1–5 generate a network G which may not be a simple network. The following algorithm takes the non-simple network G generated by one of the previous algorithms and converts it to a simple one preserving BJD matrix.

Algorithm 6

Rewiring Process

1:	while Network G is not simple do
2:	Select upper node \mathbf{u} and lower node \mathbf{v} with more than one edge between them.
3:	if There is lower node \mathbf{v}' with $deg(\mathbf{v}') = deg(\mathbf{v})$ not connected to \mathbf{u}
4:	then:
5:	Find a neighbor of \mathbf{v}' which is not neighbor of \mathbf{v} : upper node \mathbf{w}' .
6:	Remove edges \mathbf{vu} and $\mathbf{v'w'}$.
7:	Add edges $\mathbf{uv}', \mathbf{vw}'$.
8:	
9:	if There is lower node \mathbf{u}' with $deg(\mathbf{u}') = deg(\mathbf{u})$ not connected to \mathbf{v}
10:	then:
11:	Find a neighbor of \mathbf{u}' which is not neighbor of \mathbf{u} : upper node \mathbf{w} .
12:	Remove edges \mathbf{vu} and $\mathbf{u}'\mathbf{w}$.
13:	Add edges u ' v , uw .
14:	else:
15:	Find upper node \mathbf{u}' with $deg(\mathbf{u}') = deg(\mathbf{u})$, a lower node \mathbf{v}' disconnected to \mathbf{u}' with $deg(\mathbf{v}') = deg(\mathbf{v})$.
16:	Find a neighbor of \mathbf{u}' which is not neighbor of \mathbf{u} : upper node \mathbf{w} .
17:	Find a neighbor of \mathbf{v}' which is not neighbor of \mathbf{v} : upper node \mathbf{w}' .
18:	Remove edges $\mathbf{uv}, \mathbf{u'w}$, and $\mathbf{w'v'}$.
19:	Add edges $\mathbf{u}\mathbf{w}, \mathbf{w}'\mathbf{v}$, and $\mathbf{u}'\mathbf{v}'$.
20:	end if
21:	end while



Fig. 1.

In case 1, the rewiring with 3 swaps: node **u** is connected to node **v** two times, there are nodes **u**' with the same degree as **u**, and **v**' with the same degree as **v** which are not connected. There are nodes **w** (neighbor of **u**' but not a neighbor of **u**), and **w**' (neighbor of **v**' but not a neighbor of **v**). We remove edges **uv**, **u**'**w** and **v**'**w**', add edges **uw**, **vw**', and **u**'**v**'.





In case 2, the rewiring with 2 swaps: \mathbf{v}' is not neighbor of \mathbf{u} and has a neighbor like \mathbf{w}' which is not neighbor of \mathbf{v} , we remove edges \mathbf{uv} and $\mathbf{w}'\mathbf{v}'$ and add edges \mathbf{uv}' and $\mathbf{w}'\mathbf{v}$.



Fig. 3.

Left: a network generated by B2K algorithm $S_{max}ND_{min}$. Right: real Southern women network. Circle nodes on the left are women and square nodes on the right are events; nodes in each partition are listed down in descending degree.





The Romance contact network at Jefferson High [3?] consists of a single large connected component and several smaller romance groups.



Fig. 5.

Bar plot of distribution of properties of 10,000 generated networks using B2K algorithms. The different distributions result from the algorithms sampling the feasibly space differently.



(c) Romance Network (28 nodes, 288 edges)

(d) Random Network (192 nodes, 194 edges),

Fig. 6.

The structure of the giant components of Romance network and the B2K generated network are similar. The configuration model and Havel-Hakimi algorithms have the same degree distributions, but do not capture this property.



Fig. 7.

Distribution of the clustering coefficients for the ensemble of networks with the same B2K matrix as the Malaria network data. Different algorithms provide different distributions, which means the algorithms do not sample the space of feasible networks in the same way, and that the joint-degree distribution is not enough to capture clustering coefficient of Malaria data. The ND_{max} and ED_{max} B2K algorithms tend to produce networks with larger clustering coefficients than the S_{max}ND_{min} and *RE* algorithms.

Author Manuscript

Table 1

Properties of real and random Southern women network, all the properties but clustering coefficients are well estimated by all the B2K algorithms.

	Real Network	RE	ED _{max}	ND _{max}	$S_{max}ND_{min}$
<nc></nc>	-	-	-	-	1
SD(Nc)	I	0	0	0	0
<sg></sg>	32	32	32	32	32
SD(Sg)	I	0	0	0	0
<ci></ci>	0.32	0.273	0.277	0.275	0.272
SD(CI)	I	0.004	0.005	0.005	0.004
<rc></rc>	0.958	0.933	0.932	0.932	0.933
SD(Rc)	I	0.01	0.011	0.01	0.009

Table 2

Average of the properties for Southern women network and 10,000 networks generated by each of the *B1K* and *B2K* algorithms

All the algorithms could estimate Nc and Sg, however B2K algorithms could better estimate Rc. None of the algorithms is successful to estimate Cl.

Network Model	< <i>Nc</i> >	$\langle Sg \rangle$	< <i>Cl</i> >	< <i>Rc</i> >
Southern Women Network	1	32	0.32	0.958
RE	1	32	0.273	0.933
ED _{max}	1	32	0.277	0.932
ND _{max}	1	32	0.275	0.932
$S_{max}ND_{min}$	1	32	0.272	0.933
Configuration Model	1	32	0.222	0.781
Havel-Hakimi Network	1	32	0.341	0.861
Alternative Havel-Hakimi	1	32	0.291	0.916
Reverse Havel-Hakimi	1	32	0.477	0.962

Author Manuscript

Table 3

Properties of the real and randomly generated Romance networks. Note that the giant component size Sg of the real network is larger than the average size of the giant component in the randomly generated networks.

	Real Network	RE	\mathbf{ED}_{\max}	ND _{max}	$\mathrm{S}_{\mathrm{max}}\mathrm{ND}_{\mathrm{min}}$
<nc></nc>	101	102.59	110.09	109.47	99.527
SD(Nc)	I	1.791	2.51	2.499	0.71
<sg>></sg>	287	217.57	256.35	244.79	87.06
SD(Sg)	I	43.90	19.03	21.88	31.449
<ci></ci>	0.339	0.334	0.364	0.362	0.301
SD(CI)	I	0.005	0.005	0.006	0.003
<rc></rc>	0.004	0.004	0.007	0.012	0.001
SD(Rc)	I	0.006	0.007	0.009	0.003

Table 4

Average of the properties for Romance network and 10,000 networks generated by each of the *B1K* and *B2K* algorithms

The network metrics B2K algorithms are in better agreement with the metrics for real Romance network than the existing B1K algorithms. This is expected since they use more information. Note that the $S_{max}ND_{min}$ accurately captures *Cl*, and *Rc* in this low average degree network, but not *Sg*.

Network Model	< <i>Nc</i> >	< Sg >	< <i>Cl</i> >	< Rc >
Real Romance Network	101	287	0.339	0.004
RE	102.59	217.57	0.334	0.004
ED _{max}	110.09	256.35	0.364	0.007
ND _{max}	109.47	244.79	0.362	0.012
S _{max} ND _{min}	99.527	87.06	0.301	0.001
Configuration Model	103.21	203.37	0.343	0.00
Havel-Hakimi Network	178	82	0.184	0.505
Alternative Havel-Hakimi	120	50	0.383	0.068
Reverse Havel-Hakimi	132	9	0.821	0.564

Author Manuscript

Table 5

Properties of real and random Malaria network. Note that the average properties of the randomly generated networks are significantly different than the original Malaria network, although they have the same BJD matrix. This may indicate that additional properties, beyond the B2K conditions, may be needed to generate an ensemble of similar networks when the network has a high average degree.

	Real Network	RE	$\mathrm{ED}_{\mathrm{max}}$	ND _{max}	$\mathbf{S}_{\max} \mathbf{N} \mathbf{D}_{\min}$
<nc></nc>	1	1.012	1.357	1.219	1.004
SD(Nc)	I	0.107	0.509	0.427	0.063
<sg></sg>	1103	1102.96	1101.89	1102.30	1102.99
SD(Sg)	I	0.3221	1.621	1.411	0.189
<ci></ci>	0.227	0.135	0.146	0.151	0.134
SD(CI)	I	0.001	0.001	0.001	0.00
<rc></rc>	0.724	0.169	0.181	0.169	0.174
SD(Rc)	I	0.007	0.008	0.008	0.008

Table 6

Average of the properties for Malaria network and 3000 networks generated by each of the *B1K* and *B2K* algorithms

Note that the none of the algorithms estimates the network *Cl* and *Rc* accurately in this high average degree network.

Network Model	< <i>Nc</i> >	$\langle Sg \rangle$	< <i>Cl</i> >	< Rc >
Malaria Network	1	1103	0.227	0.724
RE	1.012	1102.96	0.135	0.169
ED _{max}	1.357	1101.89	0.146	0.181
ND _{max}	1.219	1102.3	0.151	0.169
$S_{max}ND_{min}$	1.004	1102.99	0.134	0.174
Configuration Model	1	1103	0.138	0.159
Havel-Hakimi Network	8	735	0.401	0.882
Alternative Havel-Hakimi	1	1103	0.191	0.413
Reverse Havel-Hakimi	10	699	0.71	0.966