# GLFR: A Generalized LFR Benchmark for Testing Community Detection Algorithms

Ba-Dung Le, Hung Nguyen, Hong Shen and Nickolas Falkner
School of Computer Science
The University of Adelaide
Adelaide, SA 5005, Australia
Email: {badung.le,hung.nguyen,hong.shen,nickolas.falkner}@adelaide.edu.au

***Abstract*—Comparisons between community detection methods are mostly based on their accuracies in recovering the built-in community structure in artificial benchmark networks. Currently used community detection benchmarks assign a fixed fraction of inter-community links, referred to as the mixing fraction, for every community in the same network. We first show in this paper that the variation in community mixing fractions has different impacts on the performances of different community detection methods that could change the decision to select a particular detecting algorithm. To comprehensively compare community detection methods, we therefore need a benchmark that generates heterogeneous community mixing fractions, which is not currently available. We address this gap by generalizing the state-of-the-art Lancichinetti-Fortunato-Radicchi benchmark to generate networks with heterogeneous community mixing fractions. Using our new benchmark, we can quantify the impact of the variation in community mixing fractions on existing community detection methods and re-evaluate the performance of the detecting algorithms as a function of the heterogeneity among the mixing fractions. Furthermore, we show that the heterogeneous community mixing tests using our generalized benchmark reflect better the performance that would be expected on real networks than the homogeneous community mixing tests using the original benchmark.**

## I. Introduction

Complex networks, such as social networks and biological networks, represent complex systems in the real world. These networks usually consist of groups of entities, also referred to as communities, with dense connections between members in the same group and sparse connections between members in different groups [13]. Identifying communities in complex networks is useful for many real-world applications such as epidemic outbreak prevention [19] and viral marketing [33].

Researchers from different disciplines have investigated a wide variety of community detection methods over the past decade [11]. Each class of the detecting algorithms generally has a distinct quality measurement of community structure [28]. Evaluating community detection methods in a unified framework is a challenging task [12]. The performance of community detection methods can be evaluated on real-world networks with known communities [26] but the number of such real networks is limited and the network sizes are often small.

Computer-generated networks [16], [7], [32], [13], [9], [18], [1], [20] can resemble different classes of real networks with community structure by tuning the network parameters. Comparisons between community detection methods are therefore mostly based on their accuracies in recovering the built-in community structure in artificial benchmark networks [21]. Currently used community detection benchmarks assign a fixed fraction of inter-community links, referred to as the *mixing fraction*, for every community in the same network. What have not been investigated are whether the variation in community mixing fractions affects symmetrically the performances of existing community detection algorithms, and more importantly whether evaluating community detection algorithms on existing benchmarks with fixed equal community mixing fractions reflect the performance that would be expected on real networks. We address both of these questions in this work.

We first show that the variation in community mixing fractions has different impacts on the performances of different community detection methods that could change the decision to select a particular detecting algorithm. To carefully evaluate the performance of community detection methods, we therefore need a benchmark that produces communities with heterogeneous mixing fractions, which is not currently available. We address this gap by generalizing the state-of-the-art Lancichinetti-Fortunato-Radicchi (LFR) benchmark [20] to account for the heterogeneity of community mixing fractions. The new benchmark allows us to quantify the impact of the variation in community mixing fractions on existing community detection methods and re-evaluate the performance of the detecting algorithms as a function of the heterogeneity among the mixing fractions. Experimental results show that the heterogeneity community mixing tests using our generalized benchmark reflect more accurately the performance that would be expected on real networks than the homogeneous community mixing tests using the original benchmark.

## II. Related work

There are now several benchmarks to evaluate community detection methods. These benchmarks differ in the generating models they use. The stochastic blockmodel [16] is a general class of random graph models with nodes divided into communities. Links are places between pairs of nodes with probabilities depending only on community memberships of the nodes. The *planted l-partition* model [7] is a realization

of the stochastic blockmodel with communities of the same size. Nodes in the same community have a probability of connection $p_{in}$, and nodes in different communities have a probability of connection $p_{out}$, with $p_{out} < p_{in}$. The Relaxed Caveman (RC) graph model [32] can be seen as a special realization of the planted l-partition model with communities formed by connecting a set of disconnected cliques. The connection probabilities $p_{in}$ and $p_{out}$ are proportional to each other and dependent on community sizes. The Girvan-Newman (GN) benchmark [13] is a specific realization of the planted l-partition model with four communities of size 32 and average node degree 16. The stochastic blockmodel based benchmarks have essentially the same degree for nodes in the same community. This is unrealistic as real networks are observed to have power law distributions in node degree and community size [2].

The degree-corrected stochastic blockmodel [18] extends the stochastic blockmodel by defining the connection probabilities of nodes based on community memberships and degrees of the nodes. This allows the variation in node degrees within communities. However, the degree-corrected stochastic blockmodel fails to resemble real networks because the link generation process of the model often produces a large number of nodes with degree zero [18]. The Lancichinetti-Fortunato-Radicchi (LFR) benchmark [20] accounts for the heterogeneity of node degrees and community sizes and is the foundation for our work in this paper. We summarize the input of the LFR benchmark in Table I and the steps of the LFR algorithm in Algorithm 1 as these details are crucial to understand our extension of the LFR benchmark.

The LFR benchmark is the most widely used benchmark as it produces networks with realistic community structure [11]. One of the main limitations of the LFR benchmark is that nodes in the LFR benchmark are assigned the same fraction of external links. The main contribution of this work is a generalization of the LFR benchmark, named GLFR benchmark, that accounts for the heterogeneity of the mixing fractions of communities. Using this generalized model we show that: (i) the performances of existing community detection methods respond asymmetrically to the variation in community mixing fractions, and (ii) the homogeneous community mixing tests conducted on benchmarks is less likely to reflect the performance that would be expected on real networks than the heterogeneous community mixing tests.

TABLE I
NETWORK PARAMETERS OF THE LFR BENCHMARK.

| Parameter | Description |
|---|---|
| $N$ | Number of nodes |
| $< k >$ | Average degree of nodes |
| $k_{max}$ | Maximum degree of nodes |
| $s_{min}$ | Minimum size of communities |
| $s_{max}$ | Maximum size of communities |
| $\gamma$ | Minus power-law exponent for node degree sequence |
| $\beta$ | Minus power-law exponent for community size sequence |
| $\mu$ | Mixing parameter |

**Algorithm 1** Steps in the LFR Algorithm [20]

1: Generate the nodes and the communities using $N$, $s_{min}$ , $s_{max}$, $\beta$
2: For each node, determine the node degree from the three parameters $< k >$, $k_{max}$, $\gamma$
3: For each node, determine the numbers of external links and internal links from $\mu$
4: For each node, create the edges between the node and the external nodes and internal nodes (randomly selected) to match the numbers of internal links and external links in step 3.

## III. GENERALIZED LFR BENCHMARK

Our main extension of the LFR benchmark is to replace the single mixing parameters $\mu$ in Table I with a set of different mixing parameters $\{\mu_1, \ldots, \mu_m\}$ for community $1, \ldots, m$ in the benchmark network. For a community $c$ with $\mu_c$ mixing fraction, nodes in community $c$ have the same fraction $\mu_c$ of external links and the same fraction $1 - \mu_c$ of internal links. Therefore, the mixing fraction of a community is also the fraction of external links for every node in the community. There are several requirements that $\mu_1, \ldots, \mu_m$ need to meet for them to be valid mixing fractions in our generalized LFR model. We list and discuss them below.

**Req. 1 (Average Mixing Fraction):** We would like the average mixing fraction to remain as $\mu$ so that our model reduces to the LFR benchmark as a special case. For this purpose, the mixing fractions of communities are taken from a uniform distribution in the range of $\mu - \Delta_\mu$ to $\mu + \Delta_\mu$.

**Req. 2 (Well-defined Community):** The mixing fraction of a community makes sense only if the community is well defined. The condition for the existence of community $c$ is [21]

$$p_i^{in} > p_i^{out}$$

for every node $i$ in the community, where $p_i^{in}$ is the connection probability between node $i$ and other nodes in community $c$, and $p_i^{out}$ is the connection probability between node $i$ and nodes outside community $c$. Let $k_i$, $k_i^{int}$ and $k_i^{ext}$ be the total degree, the internal degree and the external degree of node $i$ respectively. Let $\kappa_c$ is the total degree of nodes in community $c$, and $\kappa$ is the total degree of nodes in the whole network. The following lemma gives the condition for $\mu_c$ to be a valid mixing parameter.

**Lemma 1.** *For a community $c$ with a mixing parameter $\mu_c$, total node degree $\kappa$ and total internal node degree $\kappa_c$ to be well-defined, the condition on $\mu_c$ is*

$$\mu_c < \frac{\kappa - \kappa_c}{\kappa}$$

*Proof.* We start with the standard definitions for $p_i^{in}$ and $p_i^{out}$ in [21]

$$p_i^{in} = \frac{k_i^{int}}{\kappa_c}$$

and

$$p_i^{out} = \frac{k_i^{ext}}{\kappa - \kappa_c},$$

The condition $p_i^{in} > p_i^{out}$ becomes

$$\frac{k_i^{int}}{\kappa_c} > \frac{k_i^{ext}}{\kappa - \kappa_c}. \quad (1)$$

As $\mu_c$ is the fraction of external links of node $i$, we have

$$k_i^{int} = (1 - \mu_c)k_i \quad (2)$$

and

$$k_i^{ext} = \mu_c k_i \quad (3)$$

Substituting (2) and (3) into (1), the condition $p_i^{in} > p_i^{out}$ is equivalent to

$$\frac{(1-\mu_c)}{\kappa_c} > \frac{\mu_c}{\kappa - \kappa_c} \Rightarrow \mu_c < \frac{\kappa - \kappa_c}{\kappa}$$

□

As shown in Lemma 1, a community $c$ exists only if the mixing fraction $\mu_c$ is smaller than the ratio between the total degree of nodes outside the community and the total degree of nodes in the network. This ratio is also the maximum value that can be assigned to the mixing fraction $\mu_c$. As different communities may have different $\kappa_c$, they will have different upper-bound on $\mu_c$. For the sake of simplicity, in our GLFR benchmark we use the same maximum valid value $\mu_{max}$ for the mixing fractions of every communities in the network, which is defined as

$$\mu_{max} = \frac{\kappa - \kappa_{max}}{\kappa},$$

where $\kappa_{max}$ is the largest total degree of nodes in a community in the network

**Req. 3 (Non-negative Mixing Fraction):** The mixing fractions $\mu_c > 0$ for all communities $c$ to ensure the connectivity of the communities. This is a trivial condition but is crucial for generating a valid network. In our GLFR benchmark, we use a common lower bound $\mu_{min} = 0.025$ for all communities.

Combining **Req. 1**, **Req. 2** and **Req. 3**, the mixing fractions of communities in the network are generated from a uniform distribution in the range $[\max(\mu_{min}, \mu - \Delta_\mu), \min(\mu_{max}, \mu + \Delta_\mu)]$ with the average $\mu$ of the mixing fractions. Our GLFR benchmark takes in an additional parameter $\Delta_\mu$, named *mixing heterogeneity parameter*, in addition to those in Table I. The GLFR algorithm is presented in Algorithm 2. The main differences between Algorithm 2 and Algorithm 1 are in steps 3, 4 and 5 where different community mixing fractions are used in GLFR.

The proposed generalization of the LFR benchmark allows us to control the heterogeneity of the mixing fractions of communities by the parameter $\Delta_\mu$. For $\Delta_\mu = 0$, all the mixing fractions of communities are fixed by $\mu$ and the generated graphs are the same as the ones produced by the LFR benchmark. The greater $\Delta_\mu$, the more diverse are the mixing fractions of communities. The ambiguity of community structure is mainly controlled by the parameter $\mu$. A network

**Algorithm 2** Steps in our GLFR Algorithm

1: Generate the nodes and the communities using $N$, $s_{min}$ , $s_{max}$, $\beta$
2: For each node, determine the node degree from the three parameters $< k >$, $k_{max}$, $\gamma$
3: For each community $c$, determine the mixing parameter $\mu_c$ from a uniform distribution on the support $[\max(\mu_{min}, \mu - \Delta_\mu), \min(\mu_{max}, \mu + \Delta_\mu)]$ with the average $\mu$ of the mixing parameters.
4: For each node in community $c$, determine the numbers of external links and internal links from $\mu_c$
5: For each node, create the edges between the node and the external nodes and internal nodes (randomly selected) to match the numbers of internal links and external links in step 4.

with larger $\mu$ generally has more links, on average, between communities than with smaller $\mu$, and therefore, the built-in communities with larger $\mu$ are often harder to detect than with smaller $\mu$.

Fig. 1 illustrates the distribution of the mixing fractions of communities in the networks with $N = 1000$ and $\mu = 0.5$ for different $\Delta\mu$. We observe that our extension to the LFR bench-

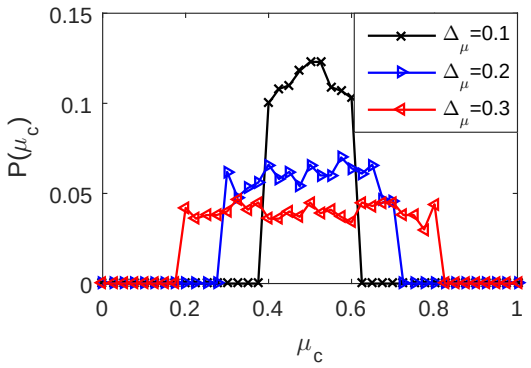



Fig. 1. Distribution of the mixing fractions of communities in networks with $N = 1000$ and $\mu = 0.5$.

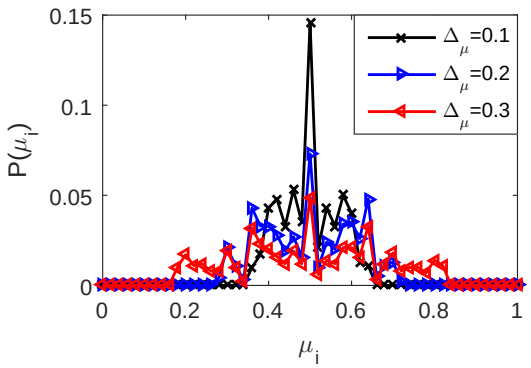



Fig. 2. Distribution of the fractions of external links of nodes in the networks with $N = 1000$ and $\mu = 0.5$.

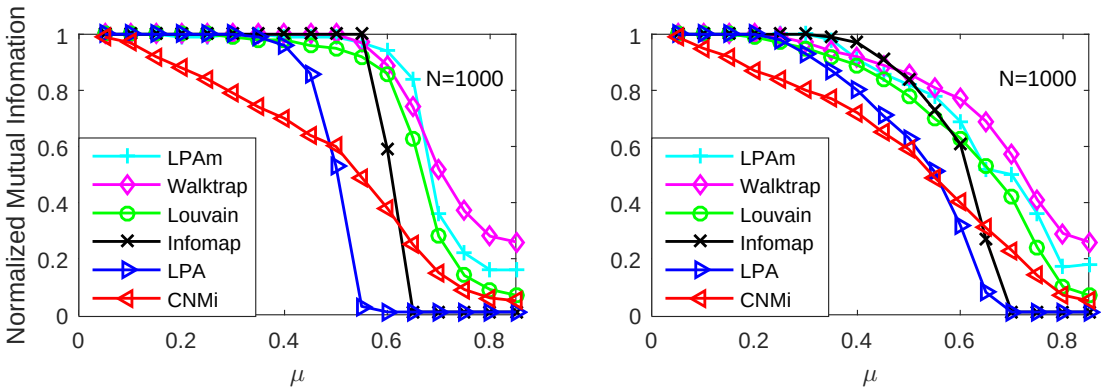
mark generates a rich heterogeneity among the community mixing fractions. The fraction of external links of every node in a community is expected to be assigned the mixing fraction of the community. However, since the number of the possible values of the fraction of external links of a node is limited and dependent on the node degree, the fraction of external links of a node is assigned the possible value closest to the mixing fraction of its associated community. The distribution of the fractions of external links of nodes in the networks with $N = 1000$ and $\mu = 0.5$ is demonstrated in Fig. 2. Again we observe a large variety of fractions of external links among the nodes. Note that the original LFR benchmark provides only a single mixing value and corresponds to the case where $\Delta\mu = 0$ in our generalized LFR model.

## IV. Applications of GLFR

We present in this section two case studies of the possible applications of our GLFR benchmark. There are potentially many more applications of GLFR that are not possible with the original LFR benchmark. We plan to explore these applications in our future works.

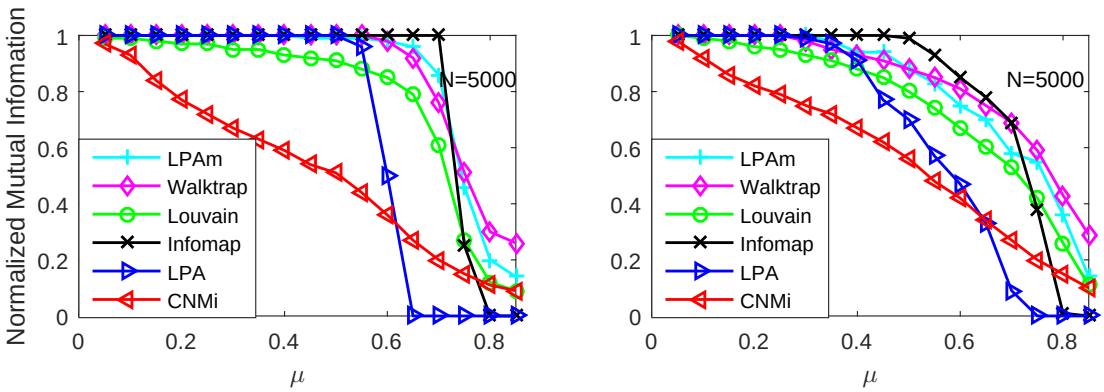
### *A. Effect of variation in community mixing fractions on community detection methods*

Our first application of GLFR is to quantify the impact of the variation in community mixing fractions on the performance of existing community detection methods. For the demonstration purpose, we test the following detection algorithms on the LFR benchmark and the GLFR benchmark. The tested algorithms whose source codes, except one, are publicly available [8] include Label Propagation Algorithm (LPA) [29], the modularity-specialized label propagation algorithm (LPAm) [3], the improved CNM algorithm (CNMi) [31], the Louvain method [4], Infomap [30], and Walktrap [27]. We use two different network sizes $N = 1000$ and $N = 5000$ representing small networks and large networks respectively to generate the synthetic networks. The parameter $\Delta_\mu$ of the GLFR benchmark is set to 0.3 for the general range of the variation in community mixing fractions. The other network parameters include the average degree of nodes $< k >$, the maximum degree of nodes $k_{max}$, the minimum community size $s_{min}$, the maximum community size $s_{max}$, the power-law exponent for the node degree sequence $\gamma$ and the power-




(a) LFR (b) GLFR, $\Delta_\mu = 0.3$

Fig. 3. (Color online) Performances of the tested methods on the LFR benchmark (a) and the GLFR benchmark with $\Delta_\mu = 0.3$ (b) for $N = 1000$. Each data point is the average over 100 network realizations.




(a) LFR (b) GLFR, $\Delta_\mu = 0.3$

Fig. 4. (Color online) Performance of the tested methods on the LFR benchmark (a) and the GLFR benchmark with $\Delta_\mu = 0.3$ (b) for $N = 5000$. Each data point is the average over 100 network realizations.

TABLE II
PERFORMANCE (NMI) OF THE TESTED METHODS ON THE LFR BENCHMARK WITH $N = 1000$ AND THE GLFR BENCHMARK WITH $N = 1000$ AND $\Delta_\mu = 0.3$ FOR $\mu = 0.5$ AND $\mu = 0.6$

| Method | N=1000 | | | |
|---|---|---|---|---|
| | $\mu = 0.5$ | | $\mu = 0.6$ | |
| | LFR | GLFR, $\Delta_\mu = 0.3$ | LFR | GLFR, $\Delta_\mu = 0.3$ |
| LPAm | 0.99 | 0.81 | **0.94** | **0.66** |
| Walktrap | 1.00 | 0.86 | **0.89** | **0.77** |
| Louvain | 0.99 | 0.78 | 0.94 | 0.63 |
| Infomap | 1.00 | 0.84 | 0.99 | 0.61 |
| LPA | 0.88 | 0.64 | 0.01 | 0.31 |
| CNMi | 0.59 | 0.59 | 0.40 | 0.41 |

TABLE III
PERFORMANCE (NMI) OF THE TESTED METHODS ON THE LFR BENCHMARK WITH $N = 5000$ AND THE GLFR BENCHMARK WITH $N = 5000$ AND $\Delta_\mu = 0.3$ FOR $\mu = 0.5$ AND $\mu = 0.6$

| Method | N=5000 | | | |
|---|---|---|---|---|
| | $\mu = 0.5$ | | $\mu = 0.6$ | |
| | LFR | GLFR, $\Delta_\mu = 0.3$ | LFR | GLFR, $\Delta_\mu = 0.3$ |
| LPAm | 1.00 | 0.89 | 0.99 | 0.76 |
| Walktrap | 0.99 | 0.89 | 0.95 | 0.80 |
| Louvain | **0.92** | **0.81** | 0.90 | 0.67 |
| Infomap | 1.00 | 0.99 | 1.00 | 0.86 |
| LPA | **1.00** | **0.70** | 0.99 | 0.45 |
| CNMi | 0.58 | 0.56 | 0.42 | 0.41 |

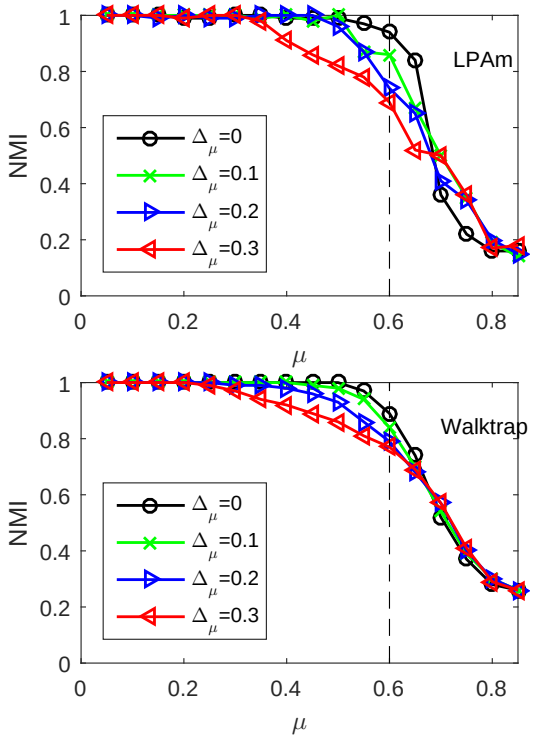



Fig. 5. (Color online) Performance of LPAm and Walktrap on the GLFR benchmark with $N = 1000$ as functions of $\mu$ and $\Delta_\mu$.

law exponent for the community size sequence $\beta$ are set as follows: $< k >= 20$, $k_{max} = 50$, $s_{min} = 20$, $s_{max} = 100$, $\gamma = -2$ and $\beta = -2$.

The performance of the tested algorithms are measured by the Normalized Mutual Information (NMI) [10], a measure of the similarity between two partitions of a network based on information theory. The normalized mutual information between partition $A$ and partition $B$ in a network is defined as

$$NMI(A,B) = \frac{-2\sum_{i=1}^{n_A}\sum_{j=1}^{n_B} n_{ij}^{AB} log(\frac{n_{ij}^{AB}.N}{n_i^A.n_j^B})}{\sum_{i=1}^{n_A} n_i^A log(\frac{n_i^A}{N}) + \sum_{j=1}^{n_B} n_j^B log(\frac{n_j^B}{N})}$$

with $n_A$ and $n_B$ are the numbers of communities in partition $A$ and partition $B$ respectively, $n_i^A$ is the number of nodes in community $i \in A$, $n_j^B$ is the number of nodes in community $j \in B$, $n_{ij}^{AB}$ is the number of the common nodes between community $i$ and community $j$, and $N$ is the number of nodes of the network.

The Normalized Mutual Information is 0 when the two compared partitions are independent from each other and 1 when the two partitions are identical.

Fig. 3 and Fig. 4 show the performance of the tested methods on the LFR benchmark and the performance on the GLFR benchmark as functions of $\mu$ for $N = 1000$ and $N = 5000$ respectively. The performance of the original CNM algorithm [6], the Louvain method and Infomap on the LFR benchmark have previously been reported in [21]. As can be seen from the figures, the performances of different tested methods change in different ways as the mixing fraction varies among communities. For example, as detailed in Table II, the performance of LPAm on the networks with $N = 1000$ and $\mu = 0.6$ decreases significantly from 0.94 on the LFR benchmark to 0.66 on the GLFR benchmark while the performance of Walktrap decreases less significantly from 0.89 to 0.77. The performance of the Louvain method on the networks with $N = 5000$ and $\mu = 0.5$ declines from 0.92 on LFR to 0.81 on GLFR while the performance of LPA drops more rapidly from 1.0 to 0.70, as detailed in Table III. It is highlighted that LPAm and Walktrap exchange their relative orders in which the methods are ranked based on their performance between the LFR benchmark and the GLFR benchmark for $N = 1000$ and $\mu = 0.6$. The Louvain method and LPA swap their relative performance ranking orders between the benchmarks for $N = 5000$ and $\mu = 0.5$. The exchange in the relative performance ranking orders of the methods could affect the decision to select a particular detecting algorithm because an algorithm with higher performance is more likely to be selected to detect communities than an algorithm with lower performance.

Fig. 5 shows the performance of LPAm and Walktrap on the GLFR benchmark with $N = 1000$ as a function of $\mu$ and $\Delta_\mu$. As can be seen from the figure, the performance of LPAm on the networks with $\mu = 0.6$, where the dash line is plotted, falls more rapidly than the performance of Walktrap does as $\Delta_\mu$ increases. This is likely due to the

reason that LPAm are more sensitive to the community mixing fractions greater than $0.6$ than Walktrap. Since LPAm merges nodes into communities to greedily improve network modularity, this algorithm easily gets stuck in poor local maxima for the large values of community mixing fractions. This is the local maxima problem of LPAm as shown in [23], [22]. Whereas, Walktrap is less affected by the local maxima problem since the algorithm iteratively merges nodes into communities based on the similarities among nodes and chooses the network partition with highest modularity obtained during the community merging process as the community structure. It is reminded that LFR is the specific realization of GLFR with $\Delta\mu = 0$. This explains why LPAm and Walktrap exchange their relative performance ranking orders between the benchmarks for $N = 1000$ and $\mu = 0.6$. The exchange of the relative performance ranking orders of the Louvain method and LPA between the benchmarks seems also to be resulted from the different sensitivities of the methods on community mixing fractions with large values.

We have shown that the performances of many existing community detection algorithms respond asymmetrically to the variation in community mixing fractions. The heterogeneity of community mixing fractions can also re-rank the relative performance orders of the methods that could affect the decision to select a particular detecting algorithm. As the mixing fractions of communities in real networks are likely more heterogeneous than homogeneous, we need to perform the heterogeneous community mixing tests on benchmarks in order to comprehensively compare community detection methods.

### *B. Comparison between LFR and GLFR*

The key application of GLFR is to evaluate the performance of community detection methods as a function of the heterogeneity among community mixing fractions. We will show in this session that the heterogeneous community mixing tests conducted on the GLFR benchmark reflect better the performance that would be expected on real networks than the homogeneous community mixing tests conducted on the LFR benchmark.

As there is no available metric to measures the difference between the performance on benchmarks and the performance that would be expected on real networks, we present our approach to derive a metric in order to compare the evaluation using the GLFR benchmark and the evaluation using the LFR benchmark. The performance of an algorithm running on a network with known communities is measured by the normalized mutual information (NMI). For synthetic network $a$ and network $b$ with ground truth, the distance between the performances of algorithm $CD_i$ can be measured by simply looking at the absolute difference in the NMI scores:

$$d_{a,b}(CD_i) = |NMI(A_i, A) - NMI(B_i, B)|$$

where $A$ is the planted community structure in synthetic network $a$ and $B$ is the ground truth in network $b$, $A_i$ and $B_i$ are the network partitions detected by method $CD_i$ on synthetic network $a$ and network $b$ respectively.

In real networks, we do not know exactly the ground truth data to compute the $NMI(B_i, B)$ as required in $d_{a,b}$. Therefore, we use the approximate ground truth in real networks as the reference to determine the approximate distance between the performances. The approximate ground truth in a real network with strong community structure [24], where communities are well separated, can be safely obtained by finding the optimal network partition with respect to a well-known community structure quality function such as modularity [25] or the map equation [30]. Since modularity is widely used as the objective function by many optimization based algorithms to detect communities [11], we define the approximate ground truth in a real network as the network partition with highest modularity found by existing community detection algorithms. Therefore, the approximate distance between the performances of algorithm $CD_i$ on synthetic network $a$ and real network $b$ with approximate ground truth $\tilde{B}$ can be measured by

$$\tilde{d}_{a,b}(CD_i) = |NMI(A_i, A) - NMI(B_i, \tilde{B})|$$

An evaluation of algorithm $CD_i$ on a synthetic network is good if the evaluation produce similar performance for that algorithm on the synthetic network and the real network resembled by the synthetic network. The evaluation of algorithm $CD_i$ therefore should have low $\tilde{d}_{a,b}(CD_i)$. We can generalize the approximate distance for an algorithm to an approximate distance for $n$ algorithms by computing the average of the approximate distances as follows.

**Definition 1.** *Let $CD_1$, $CD_2$, .., $CD_n$ be $n$ community detection methods. The approximate distance between the performance of these methods on synthetic network $a$ and the performance that would be expected on real network $b$ is*

$$\tilde{d}_{a,b}(CD_1, .., CD_n) = \frac{\sum_{i=1}^{n} |NMI(A_i, A) - NMI(B_i, \tilde{B})|}{n},$$

*where $A$ is the planted community structure on synthetic network $a$ and $\tilde{B}$ is the approximate ground truth in network $b$, $A_i$ and $B_i$ are the network partitions detected by method $CD_i$ on synthetic network $a$ and network $b$ respectively.*

We compare the evaluation of the six algorithms examined in the previous session on the LFR benchmark and the evaluation of these algorithms on the GLFR benchmark based on the proposed metric $\tilde{d}_{a,b}$. The performance of these algorithms on several commonly studied real networks with strong community structure, implied by a high network modularity value above $0.3$ [24], are used as the references to calculate $\tilde{d}_{a,b}$. The real networks include the *C.elegans* metabolic network (C.elegans) [17], the University email network (E-mail) [15], the Jazz musicians network (Jazz) [14] and the Pretty Good Privacy web of trust social network network (PGP) [5]. For the *C.elegans* metabolic network, the University email network, and the Jazz musicians network, we calculate $\tilde{d}_{a,b}$ using the performance of the algorithms on the synthetic networks with $N = 1000$ since these real networks are of small sizes

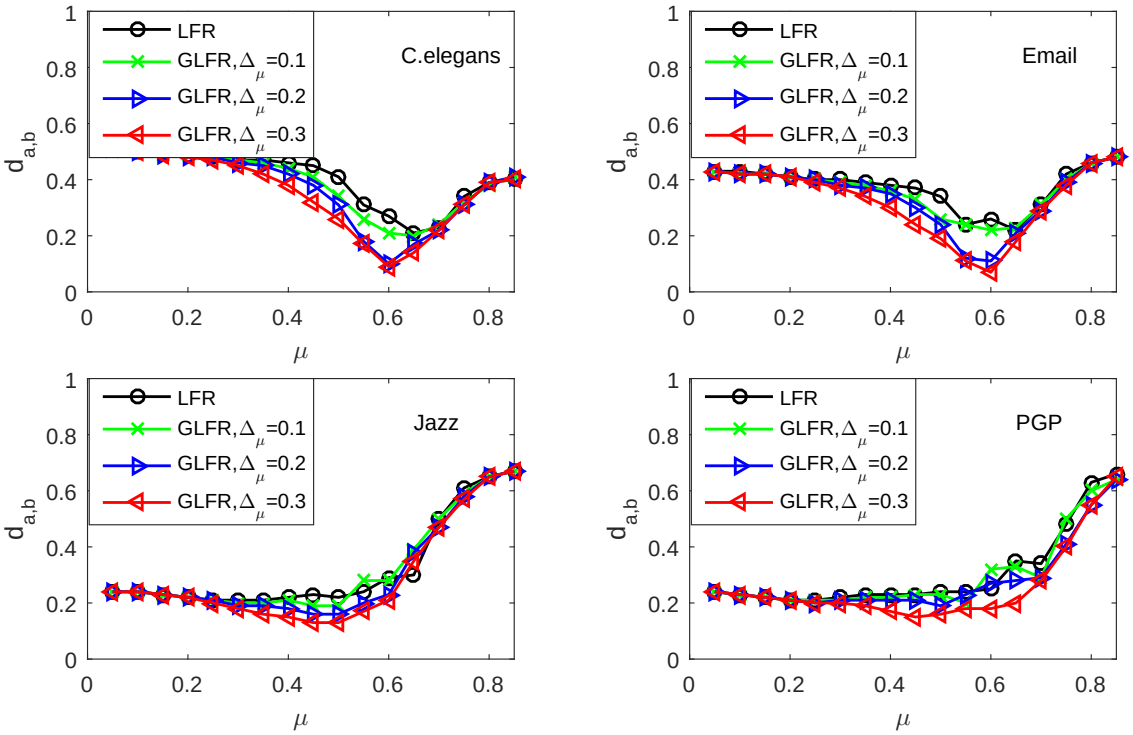



Fig. 6. (Color online) Value of $\tilde{d}_{a,b}$ of the evaluation on the LFR benchmark and the evaluation on the GLFR benchmark for four commonly studied real networks with strong community structure.

TABLE IV
THE NUMBER OF NODES $n$, THE NUMBER OF EDGES $m$, AND THE HIGHEST MODULARITY $Q_{max}$ FOUND IN THE REAL NETWORKS

| Network | n | m | $Q_{max}$ |
|---|---|---|---|
| Jazz | 198 | 2742 | 0.445 |
| *C.elegans* | 453 | 2025 | 0.451 |
| E-mail | 1133 | 5451 | 0.582 |
| PGP | 10680 | 24316 | 0.885 |

TABLE V
THE SMALLEST VALUE OF $\tilde{d}_{a,b}$ OF THE EVALUATION ON THE LFR BENCHMARK AND THE EVALUATION ON THE GLFR BENCHMARK WITH THE CORRESPONDING NETWORK PARAMETER VALUES

| Network | LFR | | GLFR | | |
|---|---|---|---|---|---|
| | smallest $\tilde{d}_{a,b}$ | $\mu$ | smallest $\tilde{d}_{a,b}$ | $\mu$ | $\Delta_\mu$ |
| Jazz | 0.21 | 0.30 | **0.13** | 0.45 | 0.3 |
| *C.elegans* | 0.21 | 0.65 | **0.09** | 0.60 | 0.3 |
| E-mail | 0.22 | 0.65 | **0.07** | 0.60 | 0.3 |
| PGP | 0.21 | 0.20 | **0.15** | 0.45 | 0.3 |

with respect to the number of nodes (approximately 1000 nodes or less). For the PGP network, we calculate $\tilde{d}_{a,b}$ using the performance of the algorithms on the synthetic networks with $N = 5000$ because this real network is of large size with regard to the number of nodes (approximately 5000 nodes or more). The approximate ground truth in the real networks we choose to be the network partition with highest modularity detected by the advanced modularity-specialized label propagation algorithm (LPAm+) [23] since this algorithm has the best performance with respect to network modularity among existing community detection algorithms as reported in [23], [22]. The network sizes and the highest modularity found by LPAm+ in the real networks are listed in Table IV.

Fig. 6 shows the value of $\tilde{d}_{a,b}$ of the evaluation on the LFR benchmark and the evaluation on the GLFR benchmark with $\Delta_\mu$ ranging from 0.1 to 0.3 as a function of $\mu$. According to the figure, the smallest value of $\tilde{d}_{a,b}$ of the evaluation on the GLFR benchmark is lower than the smallest value of $\tilde{d}_{a,b}$ of the evaluation on the LFR benchmark for all the real networks, as detailed in Table V. For example, the smallest value of $\tilde{d}_{a,b}$ of the evaluation on the GLFR benchmark for the *C.elegans* network is approximately 0.09 for $\mu = 0.6$ and $\Delta_\mu = 0.3$ while the smallest value of $\tilde{d}_{a,b}$ of the evaluation on the LFR benchmark is approximately 0.21 for $\mu = 0.65$. The smallest value of $\tilde{d}_{a,b}$ of the evaluation on the GLFR benchmark for the Jazz network is approximately 0.13 for $\mu = 0.45$ and $\Delta_\mu = 0.3$ while the smallest value of $\tilde{d}_{a,b}$ of the evaluation on the LFR benchmark is approximately 0.21 for $\mu = 0.3$. The results clearly show that the evaluation of community detection methods on the GLFR benchmark, with an appropriated benchmark parameter value-selection, resembles more closely the performance that would be expected on real networks than the evaluation on the LFR benchmark.

Evaluating community detection algorithms using the GLFR

benchmark is as easy as using the LFR benchmark. The GLFR benchmark has only an additional parameter, $\Delta_\mu$, which controls the variation in community mixing fractions with the average $\mu$. The step of generating heterogeneous community mixing fractions in the GLFR algorithm does not affect node degrees and community sizes and does not introduce more complexity into the algorithm. The GLFR benchmark therefore retains the features of real networks such as the heterogeneity in node degrees and community sizes and has the similar scalability with the LFR benchmark. Our implementation of the GLFR benchmark can be downloaded at https://github.com/dungleba/GLFR.

## V. Conclusion

Existing community detection benchmarks neglect the variation in community mixing fractions when generating synthetic networks. In this work, we have first shown that the heterogeneity in community mixing fractions has different impacts on the performances of different community detection methods that could change the decision to select a particular detecting algorithm. We therefore argue that community mixing heterogeneity should be considered seriously when evaluating community detection methods.

To generate synthetic networks with heterogeneous community mixing fractions, we have proposed a new class of benchmarks that accounts for the heterogeneity in community mixing fractions by generalizing the state-of-the-art LFR benchmark. The new benchmark allows us to quantify the impact of community mixing heterogeneity on existing community detection methods and re-evaluate the performance of the detecting algorithms as a function of the heterogeneity among community mixing fractions. We have also shown that the heterogeneous community mixing tests conducted on the new benchmark reflect better the performance of that would be expected on real networks than the homogeneous community mixing tests conducted on the original benchmark. The comparison has been made using the proposed metric $\tilde{d}_{a,b}$ that measures the distance between the performance of community detection algorithms on a benchmark and the performance that would be expected on a real network.

Further studies may explore the statistical distribution of community mixing fractions in real networks. This would help develop a more realistic benchmark with the realistic distribution of community mixing fractions. Incorporating community mixing heterogeneity into the cases of directed, weighted and overlapping benchmarks [20] could be an interesting direction for future work.

## Acknowledgment

The authors would like to thank the maintainers and contributors of the igraph packages used in this research.

## References

[1] James P Bagrow. Evaluating local community methods in networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2008(05):P05001, 2008.

[2] Albert-László Barabási and Réka Albert. Emergence of scaling in random networks. *science*, 286(5439):509–512, 1999.

[3] Michael J. Barber and John W. Clark. Detecting network communities by propagating labels under constraints. *Physical Review E*, 80(2):026129, 2009.

[4] Vincent D Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2008(10):P10008, 2008.

[5] Marián Boguñá, Romualdo Pastor-Satorras, Albert Díaz-Guilera, and Alex Arenas. Models of social networks based on social distance attachment. *Physical review E*, 70(5):056122, 2004.

[6] Aaron Clauset, M. E. J. Newman, and Cristopher Moore. Finding community structure in very large networks. *Physical Review E*, 70(6):066111, 2004.

[7] Anne Condon and Richard M Karp. Algorithms for graph partitioning on the planted partition model. In *Randomization, Approximation, and Combinatorial Optimization. Algorithms and Techniques*, pages 221–232. Springer, 1999.

[8] Gabor Csardi and Tamas Nepusz. The igraph software package for complex network research. *InterJournal, Complex Systems*, 1695(5):1–9, 2006.

[9] Leon Danon, Albert Díaz-Guilera, and Alex Arenas. The effect of size heterogeneity on community identification in complex networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2006(11):P11010, 2006.

[10] Leon Danon, Albert Diaz-Guilera, Jordi Duch, and Alex Arenas. Comparing community structure identification. *Journal of Statistical Mechanics: Theory and Experiment*, 2005(09):P09008, 2005.

[11] Santo Fortunato. Community detection in graphs. *Physics Reports*, 486(3):75–174, 2010.

[12] Santo Fortunato and Darko Hric. Community detection in networks: A user guide. *Physics Reports*, 659:1–44, 2016.

[13] M. Girvan and M. E. J. Newman. Community structure in social and biological networks. *Proceedings of the National Academy of Sciences*, 99(12):7821–7826, 2002.

[14] Pablo M Gleiser and Leon Danon. Community structure in jazz. *Advances in complex systems*, 6(04):565–573, 2003.

[15] Roger Guimera, Leon Danon, Albert Diaz-Guilera, Francesc Giralt, and Alex Arenas. Self-similar community structure in a network of human interactions. *Physical review E*, 68(6):065103, 2003.

[16] Paul W Holland, Kathryn Blackmond Laskey, and Samuel Leinhardt. Stochastic blockmodels: First steps. *Social networks*, 5(2):109–137, 1983.

[17] Hawoong Jeong, Blint Tombor, Rka Albert, Zoltan N Oltvai, and A-L Barabsi. The large-scale organization of metabolic networks. *Nature*, 407(6804):651–654, 2000.

[18] Brian Karrer and Mark EJ Newman. Stochastic blockmodels and community structure in networks. *Physical Review E*, 83(1):016107, 2011.

[19] Stephan Kitchovitch and Pietro Liò. Community structure in social networks: applications for epidemiological modelling. *PLoS ONE*, 2011.

[20] Andrea Lancichinetti and Santo Fortunato. Benchmarks for testing community detection algorithms on directed and weighted graphs with overlapping communities. *Physical Review E*, 80(1):016118, 2009.

[21] Andrea Lancichinetti and Santo Fortunato. Community detection algorithms: A comparative analysis. *Physical Review E*, 80(5):056117, 2009.

[22] Ba-Dung Le, Hung Nguyen, and Hong Shen. Community detection in networks with less significant community structure. In *Advanced Data Mining and Applications: 12th International Conference, ADMA 2016, Gold Coast, QLD, Australia, December 12-15, 2016, Proceedings 12*, pages 65–80. Springer, 2016.

[23] X. Liu and T. Murata. Advanced modularity-specialized label propagation algorithm for detecting communities in networks. *Physica A: Statistical Mechanics and its Applications*, 389(7):1493–1500, 2010.

[24] M. E. J. Newman. The structure of scientific collaboration networks. *Proceedings of the National Academy of Sciences*, 98(2):404–409, 2001.

[25] M. E. J. Newman. Modularity and community structure in networks. *Proceedings of the National Academy of Sciences*, 103(23):8577–8582, 2006.

[26] M. E. J. Newman and M. Girvan. Finding and evaluating community structure in networks. *Physical Review E*, 69(2):026113, 2004.

[27] Pascal Pons and Matthieu Latapy. Computing communities in large networks using random walks. In *Computer and Information Sciences-ISCIS 2005*, pages 284–293. Springer, 2005.

[28] Reihaneh Rabbany, Mansoureh Takaffoli, Justin Fagnan, Osmar R Zaïane, and Ricardo JGB Campello. Communities validity: methodical evaluation of community mining algorithms. *Social Network Analysis and Mining*, 3(4):1039–1062, 2013.

[29] Usha Nandini Raghavan, Réka Albert, and Soundar Kumara. Near linear time algorithm to detect community structures in large-scale networks. *Physical Review E*, 76(3):036106, 2007.

[30] Martin Rosvall and Carl T. Bergstrom. Maps of random walks on complex networks reveal community structure. *Proceedings of the National Academy of Sciences*, 105(4):1118–1123, 2008.

[31] Ken Wakita and Toshiyuki Tsurumi. Finding community structure in mega-scale social networks: [extended abstract]. In *Proceedings of the 16th international conference on World Wide Web*, Series Finding community structure in mega-scale social networks: [extended abstract], pages 1275–1276, Banff, Alberta, Canada, 2007. ACM.

[32] Duncan J Watts. *Small worlds: the dynamics of networks between order and randomness*. Princeton university press, 1999.

[33] Lilian Weng, Filippo Menczer, and Yong-Yeol Ahn. Virality prediction and community structure in social networks. *Scientific reports*, 3, 2013.