Community Detection on Graphs

Louise Allain
louise.allain@ens-paris-saclay.fr
Erkan Turan
erkan.turan@ens-paris-saclay.fr
Chiara Roverato
chiara.roverato@student-cs.fr

Abstract

This project focuses on the problem of community detection on graphs and will be conducted as follows. Beginning with a brief introduction on the subject we will introduce two reliable and efficient methods on such tasks, namely modularity and energy-based clustering algorithms. Subsequently, we will present simulated annealing, an efficient model to optimize over the set of network configurations. We will evaluate these methods on synthetic and real networks informative metrics, which will allow us to expose one of their main limitations: the resolution limit. To tackle this problem, we will finally investigate a local Pott's energy-like model.

1 Introduction

Many systems can be described as networks, composed of nodes or vertices joined together by links or edges. Such a representation enables to represent non-euclidean data while encapsulating both object attributes and their relationships. This includes a myriad of application domains which can range from social networks such as collaborative networks Rossi and Ahmed [2015], biological networks such as brains Liu et al. [2023], or technological networks such as power grids.

In the context of this project, we study the characteristic property of community structures within a graph. A community, or cluster of a graph can be defined as a sub-graph such that the inner link density within the community is higher than the average link density in the network, and this network-wide density is higher than the outer link density of the community. In practical applications, a community can correspond to people belonging to certain friend groups or football teams playing in the same league.

2 Models

Numerous detection algorithms rely on principles taking inspirations from many fields such as statistical learning, combinatorial optimization and even statistical physics. The most popular quantitative community measure is that of "modularity", originally introduced by Newman and Girvan [2004]. We will compare this classical and conventional method to a physics-inspired method, Pott's model. This latter bases its logic on spin states, that one can interprate as clusters. In this project, Chiara and Erkan will respectively work on modularity and Potts energy model and then Louise will propose an extension which bypasses certain limitations: the local Potts model.

2.1 Modularity

A significant breakthrough in community detection was made by Newman and Girvan [2004], who introduced the modularity, a quantitative measure to assess the quality of a partition of a network into communities. This measure essentially compares the number of links inside a given module (subgraph) with the expected value for a randomized graph of the same size and degree sequence. Therefore, the problem of network clustering boils down to optimizing the modularity of the network.

The modularity of a network is defined as follows:

$$Q = \sum_{s=1}^{m} \left(\frac{l_s}{L} - \left(\frac{d_s}{2L} \right)^2 \right) \tag{1}$$

where:

- m is the number of clusters or modules of the partition
- l_s is the number of links inside module s
- L is the total number of links in the network
- d_s is the total degree of the nodes in module s

In this formula, the first term is the fraction of edges of the graph inside the module and the second represents the expected fraction of edges within the community if the clustering was random.

Then, a sub-graph is a module if its impact on modularity is positive, meaning that the number of internal edges exceeds that of a random graph. It is worth noting that the modularity of a graph increases as the size of clusters or the number of well-defined communities grow. Consequently, modularity cannot be used to compare two different graphs, but is relevant when comparing diverse partitions of the same graph (Schaeffer [2007]).

By defining l_s^{out} as the number of links joining nodes from the community s to the rest of the network, we have: $d_s = l_s^{out} + 2l_s$ So the modularity can be written

as follows:

$$Q = \sum_{s=1}^{m} \left(\frac{l_s}{L} - \left(\frac{l_s}{L} + \frac{l_s^{out}}{2L} \right)^2 \right)$$

Hence, modularity is constrained in a range between -1 and 1. It is noteworthy that if $m = 1$, the two terms compensate, yielding a modularity of zero. In such cases, the observed number of edges within communities is exactly what would be expected in a random network (Fortunato [2010]). If each node constitutes a community, modularity becomes negative, as a sum of negative terms. Additionally, modularity being inferior to 1, our goal will be to maximise it, acknowledging that achieving a modularity of 1 can be unreachable on some graphs.

2.2 Potts' Model

In statistical mechanics, the Potts model is a generalization of the Ising model of interacting spin on a crystalline lattice. Reichardt and Bornholdt [2004] proposed a modified version of this model adapted for graph clustering. To do this, we assign to each node i in the graph a spin value σ_i chosen in $1, ..., q$ where q denotes the number of clusters. This spin value corresponds to the community associated to the given node. In such a context, instead of measuring clustering quality with modularity we rely on an energy defined by the following Hamiltonian.

$$H(\{\sigma\}) = -J \sum_{<i,j>} \delta_{\sigma_i,\sigma_j} + \gamma \sum_{s=1}^{q} \frac{n_s(n_s+1)}{2} \quad (2)$$

The first term is a sum over edges, $\delta_{i,j}$ denotes to the discrete Dirac function. J being a positive constant, it corresponds to a ferromagnetic interaction in the sense that this term favours an "alignement" of connected nodes along the same community, indeed connected nodes belonging to the same community lead to an energy decrease of $-J$. The second term is a sum over spin values, it involves n_s which denotes the number of nodes in a given spin/community state s. This term is parameterised by the positive constant γ and is maximum when all nodes are distributed evenly over nodes, this interaction is qualified as "anti-ferromagnetic" as it favours community "misalignement".

The choice of the γ parameter is critical to effectively detect underlying graph communities. Let's consider a network with two communities denoted by $g_1(n_1, m_1)$ and $g_2(n_2, m_2)$ interconnected by m_{12} edges. For the ground state to be composed of these two communities, γ must satisfy the following conditions:

$$H_{homogeneous} \geq H_{diverse} \quad (3)$$

Exploiting the form of the Hamiltonian yields the following:

$$-J(m_1 + m_2 + m_{12}) + \gamma \frac{(n_1+n_2)(n_1+n_2-1)}{2} \geq J(m_1+m_2) + \gamma\left(\frac{n_1(n_1-1)}{2} + \frac{n_2(n_2-1)}{2}\right) \quad (4)$$

Which yields the parameter constraint:

$$\gamma \geq J \frac{m_{12}}{n_1 n_2} \quad (5)$$

.

Comparing with equation 1 we see that when setting $J = 1$, γ exceeds a quantity corresponding to the outer link density of community $g_1(n_1, m_1)$. Thus, the γ parameter enforces that all cluster have an outer-link density smaller than γ. This observation suggests that when given a graph, if we want to satisfy the constraint 1, γ should be smaller than the average connection probability of the network $p = \frac{2M}{N(N-1)}$. With these theoretical considerations stated, the remaining task is to develop an algorithm capable of efficiently identifying the corresponding ground state.

3 Simulated Annealing

Whether it is the modularity or the Pott's energy, we now face a non-convex optimization problem which may present many local minima over a space of N^q possible graph configuration. Thus, it becomes imperative to find an efficient algorithm capable of solving this problem within a reasonable number of computations. To address this, we implement simulated annealing, an effective and precise model to approximate optimal solutions.

3.1 Algorithm

We consider the following optimization problem given by:

$$\text{minimize}_{G \in \mathbb{G}} f(G) \quad (6)$$

where $\mathbb{G}$ is the set of possible graph partitions and f some partition score, depending on the context it can be associated to modularity or Potts energy. Simulated annealing provides global optimization method, that was inspired from statistical physics (Kirkpatrick et al. [1983]), and that comes with sound theoretical guarantees (**?**). The Boltzmann distribution plays a crucial role in simulated annealing. It relates a graph partition G, the inverse temperature $\beta = \frac{1}{T}$ with $T > 0$ through:

$$P_T(G) := \exp(-\beta f(G)) \quad (7)$$

Given a randomly initialized graph partition x_1, a random configuration move Φ and parameter $\alpha < 1$, the algorithm goes as follows:

Algorithm 1 Simulated annealing

for $k = 1, ..., n_{annealing}$ **do**
  Generate a candidate configuration $y_k = \Phi(x_k)$
  Compute the acceptance probability
  $p_k = \exp(-\beta_k(f(y_k) - f(x_k))$
  if $f(y_k) - f(x_k) \leq 0$ **then**
    $x_{k+1} = y_k$
  else
    $x_{k+1} = \begin{cases} y_k & \text{with probability } p_k \\ x_k & \text{with probability } 1 - p_k \end{cases}$
  end if
  $\beta_{k+1} = \frac{\beta_k}{\alpha}$
end for

The idea of the algorithm is to pick a random configu-

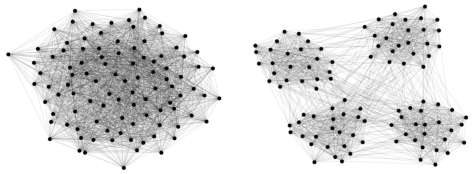
ration candidate. If it yields a decrease in the function f, it is accepted. In the event of an increase, the acceptance is performed probabilistically following the Boltzmann distribution. This treatment allows to probe short term "suboptimal" moves which may lead to long term gains by potentially escaping local minima. To reach a good estimate of the optimal configuration, we progressively decrease the temperature to promote moves which yield a decrease in f. There is some flexibility in the type of move Φ we can operate on our partitions, this is problem dependent and will be specified in the Results section.

4 Benchmarks

4.1 Naive synthetic benchmark

To test our algorithm for controlled experiments against ground truth, we first built a very simple synthetic graph inspired by Newman and Girvan [2004] as follows. We set the number of nodes and the number of clusters each containing an equal number of constituents. We take in two parameters p_{in} and p_{out} corresponding to the intra-connectivity and inter-connectivity probability. p_{in} corresponds to the probability of two nodes in the same community to form an edges whilst p_{out} corresponds to the probability of two nodes of different communities to form an edge. Figure 1 shows what this graph looks like for $p_{in} = 0.5$ and $p_{in} = 0.9$, we see that as p_{in} increases the clusters clearly get more defined.

Figure 1: Naive benchamark: on the left we have $p_{in} = 0.5$ and on the right we have $p_{in} = 0.9$



4.2 Realistic synthetic benchmark

A better way of testing our algorithms is to build a more realistic graph. To do so, we implement the algorithm proposed by Lancichinetti et al. [2008]. We aim to construct graphs that reflect the diversity of communities within the same graph. In particular, we argue that in graphs issued from real-life data, the nodes of a same graph can present a wide range of degrees and the communities might have fairly different sizes from one another. Moreover, it has been noted in the past years that real networks' degrees distribution's tail often decay as power laws. In the same fashion, it seems that the tail of the community size distribution can be asymptotically approximated by a power law Palla et al. [2005].

This leads us to a benchmark's algorithm in which the degree and the community size power laws are drawn according to power laws. The algorithm is as follows :

1. Initialization of the degree of each node according to a power law of parameter γ and of average degree k,
2. Initialization of the size of each community according to a power law of parameter β,
3. Each node is randomly assigned to a community in a way that :
   - The community size is inferior to the internal degree of the node if assigned to this community,
   - If when trying to add a node to a community, the community is full (the community size is already attained), a random node of the community is ejected.
4. The graph is rewired in a way that the internal degree of each node is roughly μ

To rewire the graph, we developed a strategy that considered the neighborhood of a node was correct if $|\mu(node) - \mu| \geq \varepsilon$ for a chosen ϵ. From this, we first rewire nodes that need to increase their internal degree by randomly selecting two nodes of the same community that require such a rewiring and rewiring them together while wiring together two of their neighbors of a different community and that need less internal edges. In a second step, we rewire only nodes that have too much internal edges by exchanging inner-community edges with nodes of a different community.

Such an algorithm enables us to generate fairly quickly numerous large graphs with a wide variety of characteristics by changing the parameters μ, γ and β.

Unfortunately, due to the size of the graphs we could not perform any tests on this benchmark.

4.3 Real Datasets

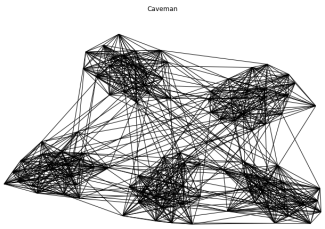
To test our algorithms on real world examples with known community structure, we use the College Football network (Girvan and Newman [2002]). It represents the game schedule of the 2000 season of Division I of the US college football league. The nodes in the network represent 115 teams, while the links represent the 613 games played in the course of a year. Community structures naturally arise from the grouping of teams into conference of around 8-12 teams each with each team playing 7 matches with teams of their own conference and 4 with the outer conference.

5 Results

Modularity and Potts model both used simulated annealing which offers flexibility on the choice of temperature schedule and types of random configuration moves. For the temperature schedule, in both cases we opted with an initial high temperature and decreased at each annealing step according to the following sequence $T_{k+1} = \alpha T_k$ with α the cooling rate ranging between 0.7 to 0.95. In the case of the standard Potts model, at each annealing step, performing N^2 random community change, where N desig-

nates the number of nodes of the graph seemed to work best. In the case of modularity optimization, the number of random cluster changes was determined through a grid search along with the other parameters. We conducted multiple experiments on synthetic data, starting with a Caveman type network as shown in Figure 1. Reichardt and Bornholdt [2004]

Figure 2: Caveman Graph



We observe the expected evolution of the energy and modularity scores through iteration steps as depicted in Figure 1.

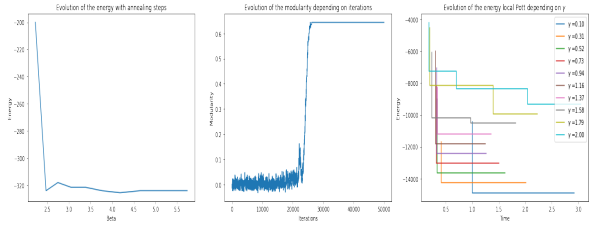


Figure 3: Energy and Modularity scores through iterations on Caveman Graph

For a more visual representation of the clustering we computed the co-appearance matrix described in Reichardt and Bornholdt [2004] which yielded matrices presenting clear community structure as shown in Figure 3.

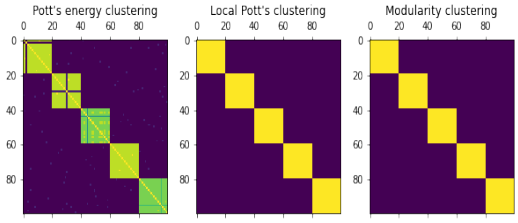


Figure 4: Co-appearance matrices for the Caveman Network

For a more quantitative analysis on performance we then performed experiments on the naive benchmark to see how our algorithm would resolve clusters which are less and less defined. To do so we computed the sensitivity and specifity of the obtained graph partitions for varying values of intra-connection probability p_{in}, for each value of p_{in} started from multiple random initialization and reported the mean and error bars. Sensitivity and specificity are benchmarked over all possible pairs of nodes. As true positive (negative) we count a pair of nodes that is in the same (a different) community by design and is classified accordingly by the algorithm. The results for all three algorithms are depicted in Figure 4 and as expected performance increases as the clusters are more defined.

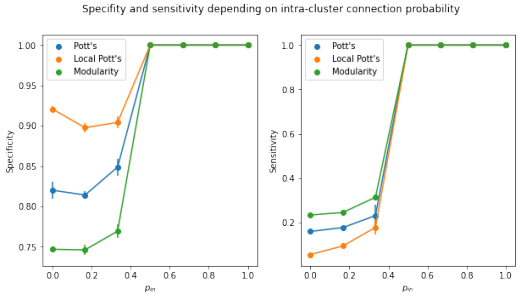


Figure 5: Evolution of specifity and sensitivity on naive benchmark as we increase p_{in} for Potts, Local Potts and Modularity

We also studied the robustness of Potts model clustering with respect to the choice of number of clusters q as shown in Figure 5, to do so we ran a single annealing step at $T \approx 0$ started from 10 randoms initializations we plotted the mean values of specifity and sensivity with their error bars. We see that performance is quite insensitive to this initialization parameter. This is quite an advantagous as we rarely have access to the number of clusters beforehand.

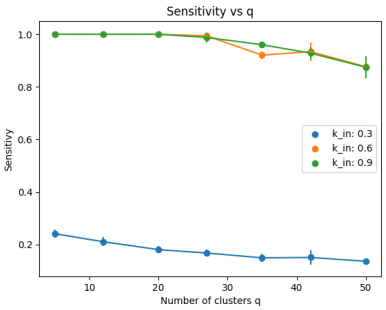


Figure 6: Co-appearance matrices for the Caveman Network

Finally, we put our algorithms to the test on a a real dataset, namely the Football dataset described in Section 4. Inspired by Reichardt and Bornholdt [2004] we tested the robustness of the clusters. Using the Potts model, We scanned through γ values going from $0.1p \leq \gamma \leq p$ and performed 10 single annealing steps at temperatures $T \approx 0$ on randomly initialized starts. We get the co-appearance matrix that we reordered according to the resulting community attribution obtained on the final step, the final matrix appears in Figure 6.

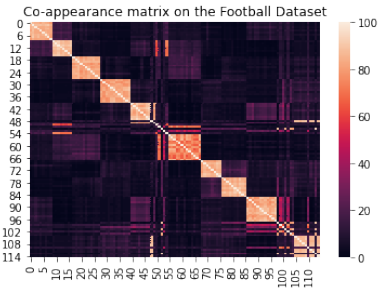


Figure 7: Co-appearance matrices resulting from Potts model on football dataset

We obtain a result comparable to Reichardt and Bornholdt [2004], we see that some clusters are divided which corresponds which is in line with the the fact that teams in a given league are more likely to play against geographically close teams. On Figure 7 we present the resulting clustering obtained using the three models.

We also performed experiments on the realistic benchmark, unfortunately due to time constraints we weren't able to produce the results in this report.

6 Limitations

Although Modularity optimization is a very effective method to detect communities in real and synthetic setting it still present some limitations. Indeed Fortunato et al. [cite] find that modularity contains an intrinsic scale which depends on the topology of the networks and that modules smaller than that scale may not be resolved.

6.1 The resolution limit

The resolution limit was thoroughly discussed by Fortunato and Barthélemy [2007] in the general setting. We'll illustrate this phenomenon on an informative example. Suppose we have a network consisting of a ring n of cliques, connected through single links. Each clique has m nodes. Suppose we have an even number of cliques. The modularity Q_{single} of the natural partition is:

$$Q_{single} = 1 - \frac{2}{m(m-1)+2} - \frac{1}{n} \tag{8}$$

Whilst the modularity Q_{pairs} consisting of a partition grouping pairs of consecutive cliques as a single clusters is:

$$Q_{pairs} = 1 - \frac{1}{m(m-1)+2} - \frac{2}{n} \tag{9}$$

The condition $Q_{single} > Q_{pairs}$ is satisfied if and only if:

$$m(m-1)+2 > n \tag{10}$$

This condition is not always satisfied, the case $m = 5$ and $n = 30$ is an example of that. This means that an efficient algorithm seeking for maximum modularity won't be able to resolve the true partition.

6.2 Consequences

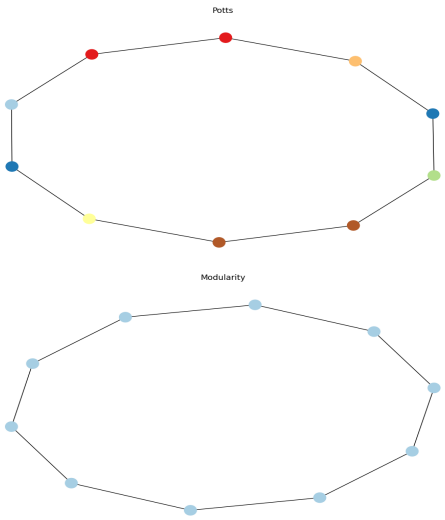


Figure 9: Resulting clusters of simulated annealing performed with Potts and modularity on a cycle

We can take as an example a simple cycle graph of 10 nodes. This graph can be seen as 5 clusters each containing a clique of 2 nodes. Even if it seems easy to cluster, simulating annealing fails to perform as shown in 9. For modularity, we remark that all the clusters are merged, which results from the previous theorical observations.

6.3 A local resolution-limit-free Potts model

To tackle the resolution limit problem, and display an algorithm that is insensitive to this phenomenon, we implement a Potts type model, both robust to noise, fast and scalable. This model-free approach aims to minimize the energy

$$\mathcal{H}(\{\sigma\}) = -\frac{1}{2}\sum_{i\neq j}(a_{i,j}A_{i,j} - \gamma b_{i,j}J_{i,j})\delta(\sigma_i,\sigma_j),$$

where A is the adjacency matrix, $J \simeq 1 - A$ with a null diagonal and a and b correspond to edge weights. Moreover, γ is a parameter that enables us to choose whether we want to concentrate more on the internal edges of a community or the missing ones, and σ and δ are as defined above. In our implementation, we consider only unweighted graphs and thus $a_{i,j} = b_{i,j} = 1$.

With this definition of the energy, the missing edges are directly penalized and communities are defined by their edge densities.

The algorithm is as follows :

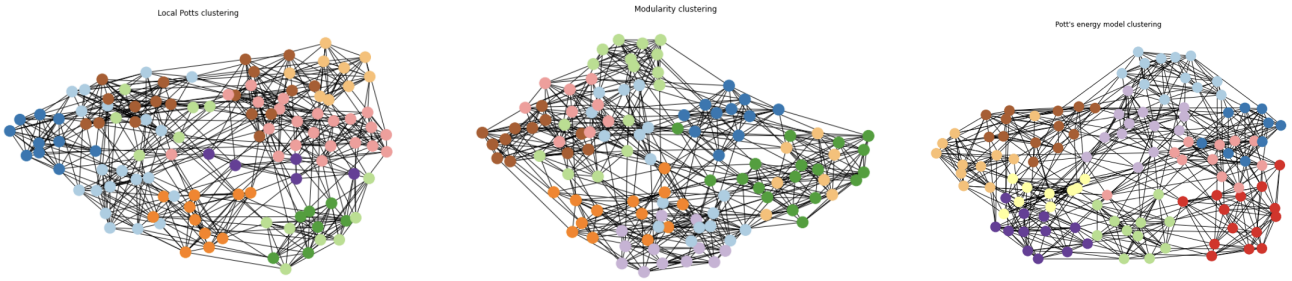


Figure 8: Resulting clustering with the three approaches

Algorithm 2 Local Potts

1: σ(node) = node
2: **while** configuration changed **do**
3: **for** node in nodes **do**
4: σ(node) = $\arg\min_s(\mathcal{H}(\{\sigma\}))$ if node is placed in community s
5: **end for**
6: **end while**
7: **for** (s, t) communities, $s \neq t$ **do**
8: **if** merge(s, t) lowers $\mathcal{H}$ **then**
9: merge(s, t)
10: **end if**
11: **end for**
12: **if** any communities were merged **then**
13: return to line 2
14: **end if**

In other words, after initializing each node a unique community, we iterate through the nodes and move them to the best neighboring community. Once no node can be moved to a community anymore, we try to merge communities, if we succeed, we get back to optimizing the node membership. The idea here is that communities won't merge even if they are very small, only the edge density within the communities will matter, thus, two communities will merge only one of their edge density p is small enough, which happens when

$$p < \frac{\gamma}{\gamma + 1}.$$

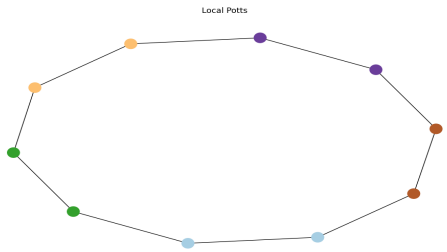


Figure 10: Resulting clusters of simulated annealing performed with Potts and modularity on a cycle

This idea can be confirmed when we try to cluster the previous cycle graph, which is done with no difficulties as show in 10.

Conclusion

In this project, we explored two prominent approaches for community detection in graphs: the modularity-based method and a physics-inspired Potts model. We implemented and optimized these algorithms using simulated annealing, a powerful optimization technique. Our experiments covered both synthetic and real-world networks, providing insights into the strengths and limitations of each method.

The modularity-based approach, with its focus on optimizing the quality of partitions, demonstrated effectiveness in various scenarios. However, we discussed its intrinsic resolution limit, which may hinder its ability to detect smaller communities in certain network structures.

To address this limitation, we proposed a local Potts model that aims to be resolution-limit-free. This model, driven by an energy function, considers both internal and missing edges, providing a more flexible perspective on community detection.

Our experiments on synthetic benchmarks and real datasets, such as the College Football network, allowed us to evaluate the performance of these algorithms. The results highlighted the trade-offs between modularity and the Potts model and emphasized the importance of considering the specific characteristics of the network when choosing a detection method.

In conclusion, community detection in graphs is a nuanced problem with various factors influencing the choice of an appropriate algorithm. The modularity-based method excels in many cases but may face challenges in the presence of a resolution limit. The local Potts model offers a promising alternative, providing a more versatile and resolution-limit-free approach. Further research and experimentation are warranted to refine and extend these models for a broader range of network structures and applications.

References

Santo Fortunato. Community detection in graphs. *Complex Networks and Systems Lagrange Laboratory, ISI Foundation*, 2010.

Santo Fortunato and Marc Barthélemy. Resolution limit in community detection. *Proceedings of the National Academy of Sciences*, 104(1):36–41, 2007. doi: 10.1073/pnas.0605965104. URL https://www.pnas.org/doi/abs/10.1073/pnas.0605965104.

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. doi: 10.1073/pnas.122653799. URL https://www.pnas.org/doi/abs/10.1073/pnas.122653799.

S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi. Optimization by simulated annealing. *Science*, 220 (4598):671–680, 1983. doi: 10.1126/science.220.4598.671. URL https://www.science.org/doi/abs/10.1126/science.220.4598.671.

Andrea Lancichinetti, Santo Fortunato, and Filippo Radicchi. Benchmark graphs for testing community detection algorithms. *Physical Review E*, 78(4), October 2008. ISSN 1550-2376. doi: 10.1103/physreve.78.046110. URL http://dx.doi.org/10.1103/PhysRevE.78.046110.

Lingwen Liu, Guangqi Wen, Peng Cao, Tianshun Hong, Jinzhu Yang, Xizhe Zhang, and Osmar R. Zaiane. Braintgl: A dynamic graph representation learning model for brain network analysis. *Computer Science and Engineering, Northeastern University*, 2023.

M. E. J. Newman and M. Girvan. Finding and evaluating community structure in networks. *Phys. Rev. E*, 69(2):026113, 2004.

Gergely Palla, Imre Derényi, Illés Farkas, and Tamás Vicsek. Uncovering the overlapping community structure of complex networks in nature and society. *Nature*, 435:814–818, 07 2005.

Jörg Reichardt and Stefan Bornholdt. Detecting fuzzy community structures in complex networks with a potts model. *Phys. Rev. Lett.*, 93:218701, Nov 2004. doi: 10.1103/PhysRevLett.93.218701. URL https://link.aps.org/doi/10.1103/PhysRevLett.93.218701.

Ryan A. Rossi and Nesreen K. Ahmed. The network data repository with interactive graph analytics and visualization. In *AAAI*, 2015. URL https://networkrepository.com.

Satu Elisa Schaeffer. Graph clustering. *Laboratory for Theoretical Computer Science, Helsinki University of Technology TKK*, 2007.