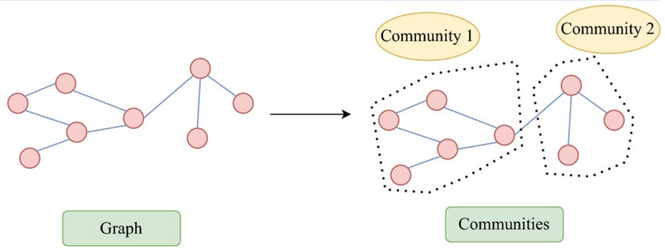
Algorithms for Community Detection: Analysis and Applications

Tony Ni*, Jake Jasmer*, Aidan Roessler*, Yang Tan*, Layla Oesper
Carleton College, Northfield, MN
* Indicates equal contribution

Paper, code, and data are available:
https://github.com/ZhanghanNi/community-detection-blue

The Community Detection Problem



Input: A graph $G = (V, E)$ and a function Q that measures the quality of a partitioning of V

Output: A partitioning of V into disjoint subsets $\mathscr{C} = \{C_1, C_2, ..., C_k\}$ s.t. $\bigcup_{i=1}^{k} C_i = V$ and $Q(\mathscr{C})$ is optimized

Our contributions:
1. Implemented three algorithms for community detection
2. Designed synthetic datasets to benchmark space and time complexity as well as performance metrics
3. Evaluated performance across three real-world community detection scenarios

Algorithms for Community Detection

1. **Girvan-Newman Algorithm**
Reveals communities by removing the "most valuable" edge (e.g. the one with the highest betweenness centrality). As the graph breaks down into pieces, the tightly knit community structure is exposed.

divisive approach

Betweenness centrality: how often an edge appears on the shortest path between pairs of nodes

2. **Louvain Algorithm**
Detects communities by iteratively merging nodes to maximize a heuristics function (e.g. modularity). Nodes first move to neighboring communities if it improves modularity, then detected communities are collapsed into single nodes, and the process repeats until no further improvement is possible.

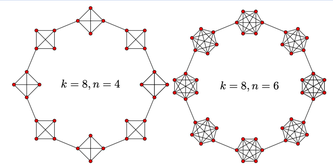
agglomerative approach

3. **Basic Variable Neighborhood Search (BVNS)**
Detects community by exploring different "neighborhoods" of solutions. It starts with an initial partition and iteratively applies three steps until no further improvement is possible:
- Shaking: Randomly perturbs the current best solution to escape local optima.
- Local search: Moves nodes between communities to improve modularity.
- Neighborhood change: Updates the best solution and adjusts the search radius.

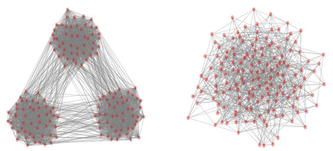
randomized approach

Synthetic Dataset Design

1. **Ring of Cliques**
A ring of cliques graph consists of k cliques, each of size n, connected by single links. Each clique forms a complete graph.

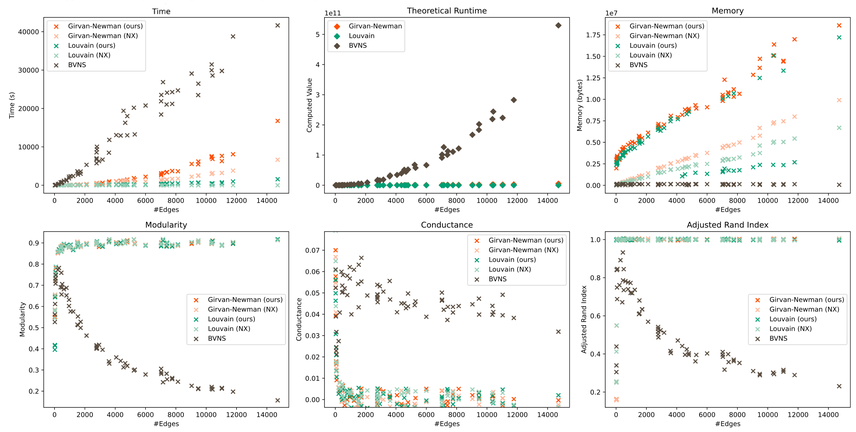


2. **Stochastic Block Model (SBM)**
This model partitions nodes into n blocks of specified sizes, connecting node pairs independently based on a probability matrix that controls intra-community density p and inter-community density q.

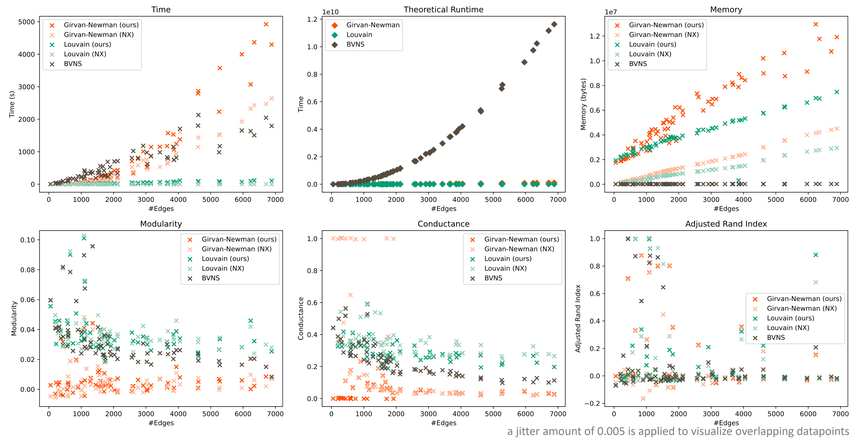


Computational Complexity and Performance

1. **Ring of Cliques (density=0.1)**



2. **Stochastic Block Model (density=0.8)**



a jitter amount of 0.005 is applied to visualize overlapping datapoints

Key observations moving from simple, sparse graphs to challenging, dense graphs:
1. runtime of Girvan-Newman eventually outgrows that of BVNS
2. peak memory usage of Girvan-Newman significantly surpasses that of Louvain
3. all algorithms struggle with dense SBM graphs, with Girvan-Newman experiencing the most significant performance decrease

Evaluation Metrics

1. **Modularity**: measures the strength of a network's partition into communities by comparing its structure to a randomized network with the same number of nodes, edges, and degree distribution.
2. **Conductance**: the fraction of edges crossing between communities relative to the total connections within the smaller community, averaged over all community pairs.
3. **Adjusted Rand Index (ARI):** the similarity between two partitions of by comparing the assignments of data points to communities.

Actual weight between nodes i & j — Expected weight between nodes i & j (probability of two half-edges being connected)

$$Q = \frac{1}{2m} \sum_{ij} \left[A_{ij} - \frac{k_i k_j}{2m} \right] \delta(c_i, c_j)$$

Total number of half-edges — 1 if nodes i & j are in the same community, 0 otherwise

higher Q means stronger community structure

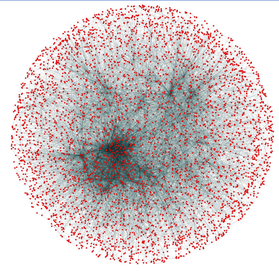
lower conductance means more separated communities

higher ARI means better alignment with ground truth

Application: Protein-Protein Interaction

Dataset: we detect communities in *Saccharomyces cerevisiae* (yeast) Protein-Protein Interaction (PPI) network, where a vertex represents a protein and an edge represents the interaction between two proteins.

Significance: community detection identifies proteins that functions together, which corresponds to protein complexes or segments of biological pathways. This can assist protein function annotation and drug target discovery.



PPI network of Yeast

Algorithm	# Communities	Modularity	Runtime (s)	Peak Memory (MB)	F-measure	Accuracy	Composite Score
Louvain (ours)	162	0.780	59.540	22.340	0.205	0.393	0.834
Louvain (NX)	114	0.953	**0.103**	**1.074**	**0.216**	**0.432**	**1.009**
Girvan-Newman (ours)	115	**0.954**	190.330	38.265	0.215	0.429	0.996
Girvan-Newman (NX)	115	**0.954**	155.027	20.874	0.215	0.429	0.996
BVNS	34	0.400	19462.016	20.026	0.000	0.191	0.337

F-measure is the harmonic average of Precision and Recall; Accuracy is the geometric average of Sensitivity and Positive Predicted Value; Composite Score is the sum of Precision, Sensitivity, and Accuracy

Louvain is the best for detecting communities in yeast PPI networks!

Conclusions

Across both synthetic dataset and real-world datasets:
1. **Louvain**: Achieves the fastest runtime, highest modularity, and greatest overlap with ground truth. The difference between the NetworkX's implementation and ours is more pronounced in real-world datasets, likely due to differences in heuristic functions used.
2. **Girvan-Newman**: Has modularity values similar to Louvain but with a slower runtime. Performs best on sparsely connected graphs.
3. **BVNS:** Performs worst in runtime, modularity, and overlap with ground truth. Since the number of optimization steps remains constant regardless of input graph size, BVNS performance decreases as graph size increases.
4. NetworkX's implementations outperform ours in runtime and peak memory usage.

Acknowledgments
We thank Professor Layla Oesper for guidance and support throughout this project, Mike Tie for computational resource support, and everyone in the Graph Problems Comps group for meaningful discussions.

References
Blondel, V. D., Guillaume, J.-L., Lambiotte, R. & Lefebvre, E. Fast unfolding of communities in large networks. *J. Stat. Mech.* **2008**, P10008 (2008).
Girvan, M. & Newman, M. E. J. Community structure in social and biological networks. *Proceedings of the National Academy of Sciences* **99**, 7821–7826 (2002).
Jovanović, D., Davidović, T., Urošević, D., Krüger, T. J. & Ramljak, D. Variable Neighborhood Search Approach to Community Detection Problem. in *Numerical Methods and Applications* (eds. Georgiev, I., Datcheva, M., Georgiev, K. & Nikolov, G.) 188–199 (Springer Nature Switzerland, Cham, 2023).
Kumar, S., Mallik, A. & Sengar, S. S. Community detection in complex networks using stacked autoencoders and crow search algorithm. *J Supercomput* **79**, 3329–3356 (2023).
Saha, S., Chatterjee, P., Basu, S., Nasipuri, M. & Plewczynski, D. FunPred 3.0: improved protein function prediction using protein interaction network. *PeerJ* **7**, e6830 (2019).
Wang, J., Cao, J., Li, W. & Wang, S. CANE: community-aware network embedding via adversarial training. *Knowl Inf Syst* **63**, 411–438 (2021).