Advanced Tools for Complex Network Analysis CNET 5052, Spring 2026, Assignment 2

Colin White

1 Bayesian Statistics and Networks

The most likely m value is 3 for both.

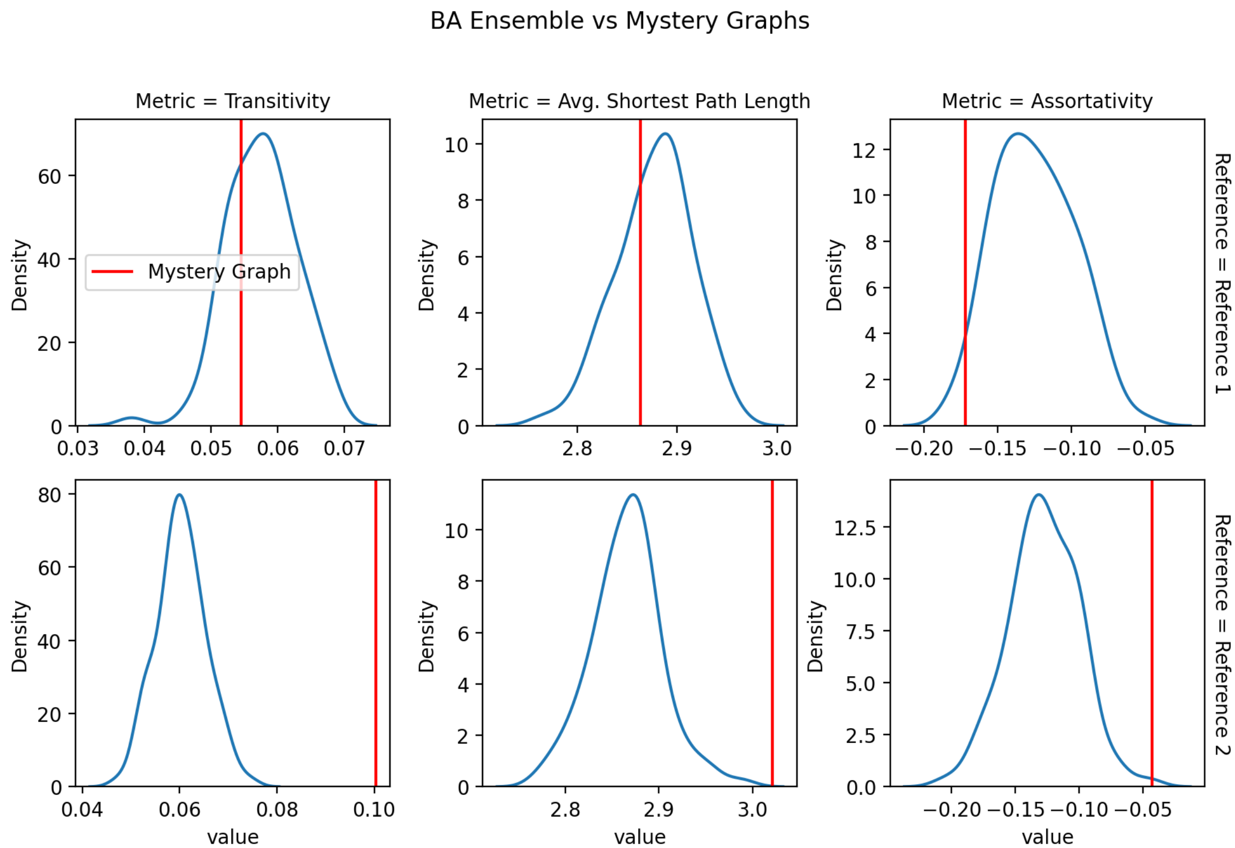

Figure 1: mystery network (red lines), compared to ensemble attributes distribution

The null-hypothesis is the mystery network(s) were created using a BA model. For mystery graph 1 (top row, figure 1) we cannot reject the null hypothesis as its attributes are consistent with the ensemble of BA models with m=3. For mystery graph 2 (bottom row, figure 1) we do reject the null hypothesis. Because all three attributes fall far outside the BA ensemble distribution, meaning it is very unlikely to have been generated by the BA model.

2 Replication: Modularity in random graphs

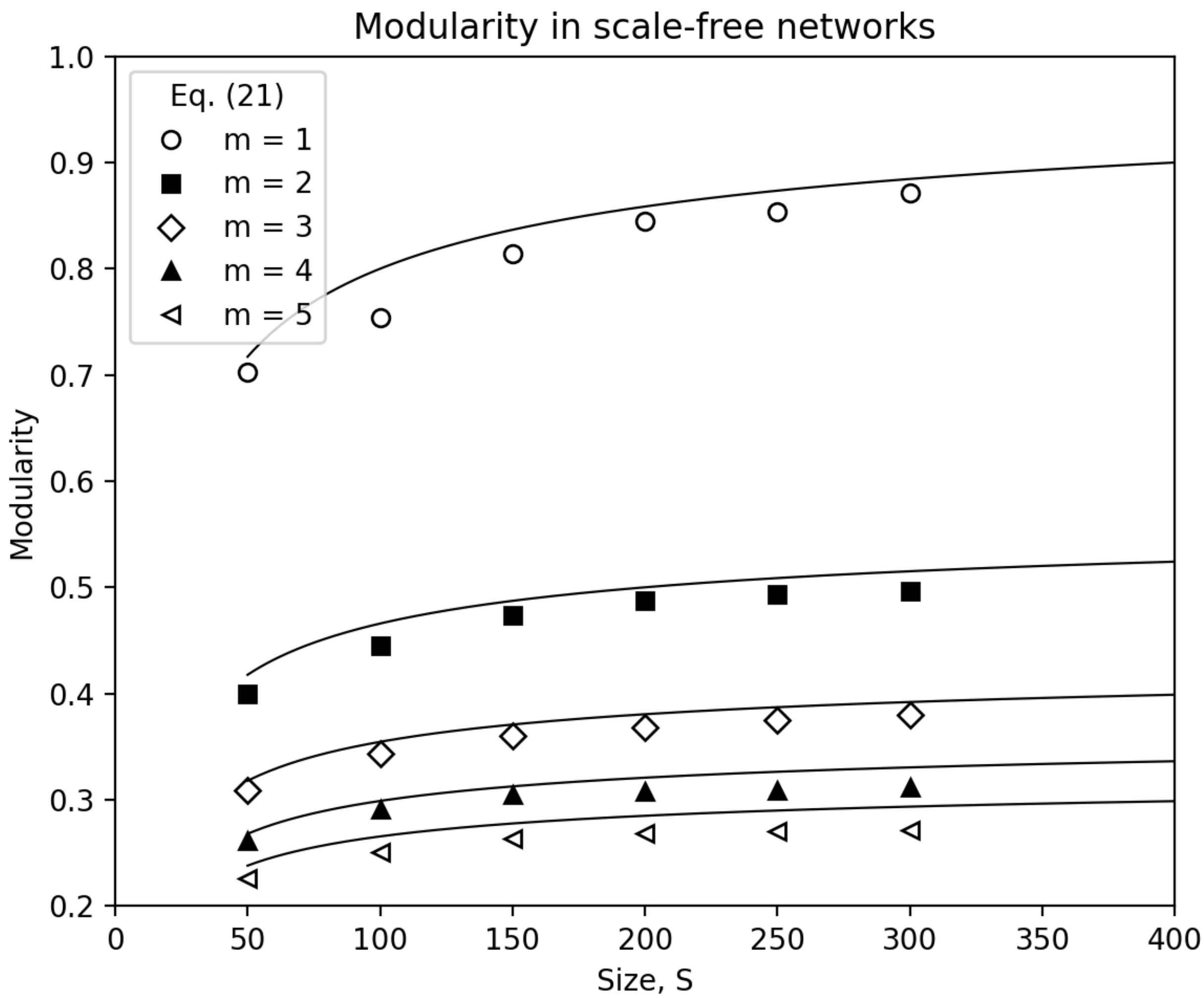

Figure 2: Replication of Figure 2 from Roger Guimerà, Marta Sales-Pardo, and Luís A. Nunes Amaral, (2004), ”fluctuations in random graphs and complex networks”,

The core finding is modularity alone is insufficient for determining community validity. Figure 2 shows that even fully random BA graphs can have high modularity, which we know must be due to probability distribution of linking. For example, with parameters m=1 and 300 nodes, modularity reaches approx 0.85. A naive interpretation could assume this means a strong community structure exists. But we made this to be a random BA graph, so strong community structure can not be the case. There are two insights we can take in the figure. First, modularity decreases as m increases. Higher m means more edges per node, making it harder for the probability mechanism to create seeming clusters. At m=5 the effect is smaller but still substantial. Second, modularity increases with network size S. This seems odd at first because it appears to imply that larger random networks are more modular. The insight we draw from this is that modularity should not be interpreted on an absolute scale, but should always be compared against an ensemble of null models of a random graph with the same size and m.

3 Exploring community detection

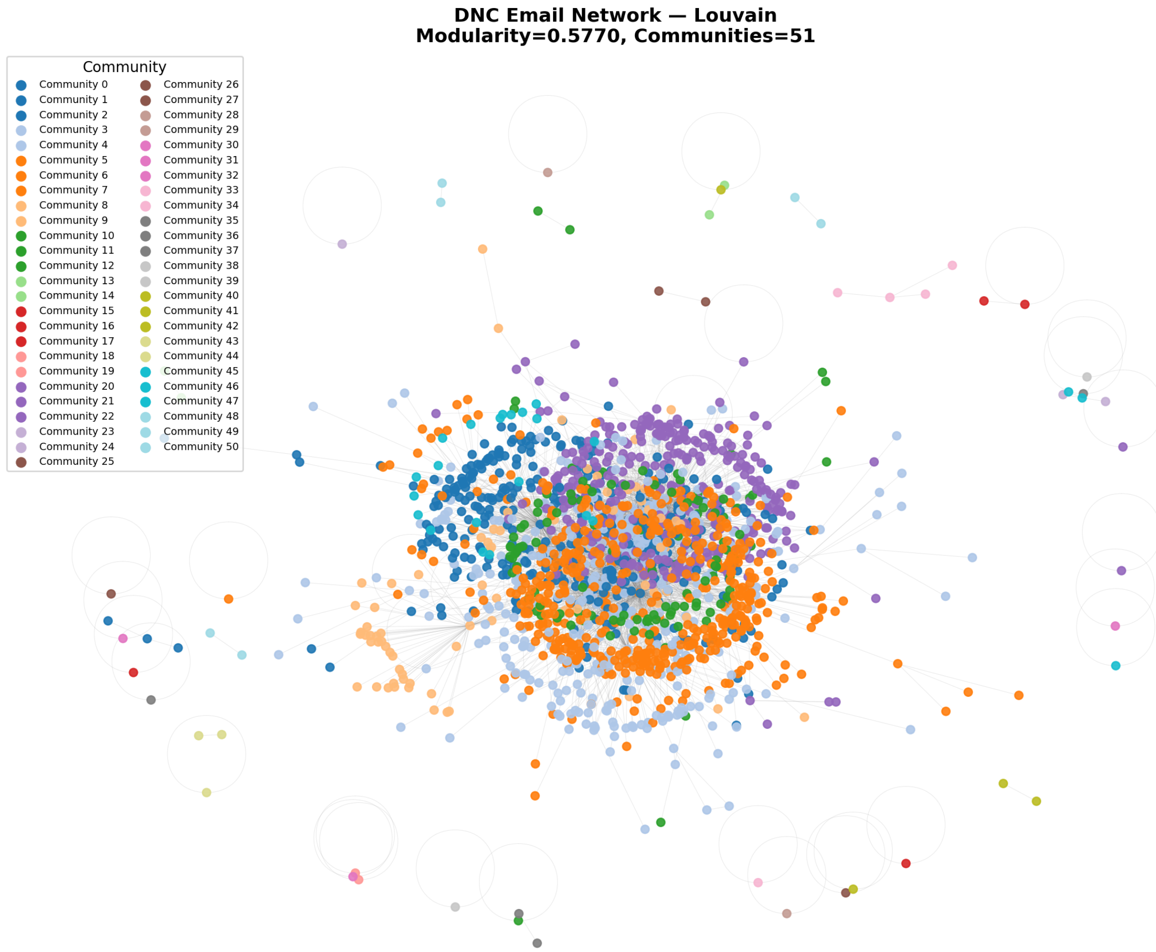

Figure 3

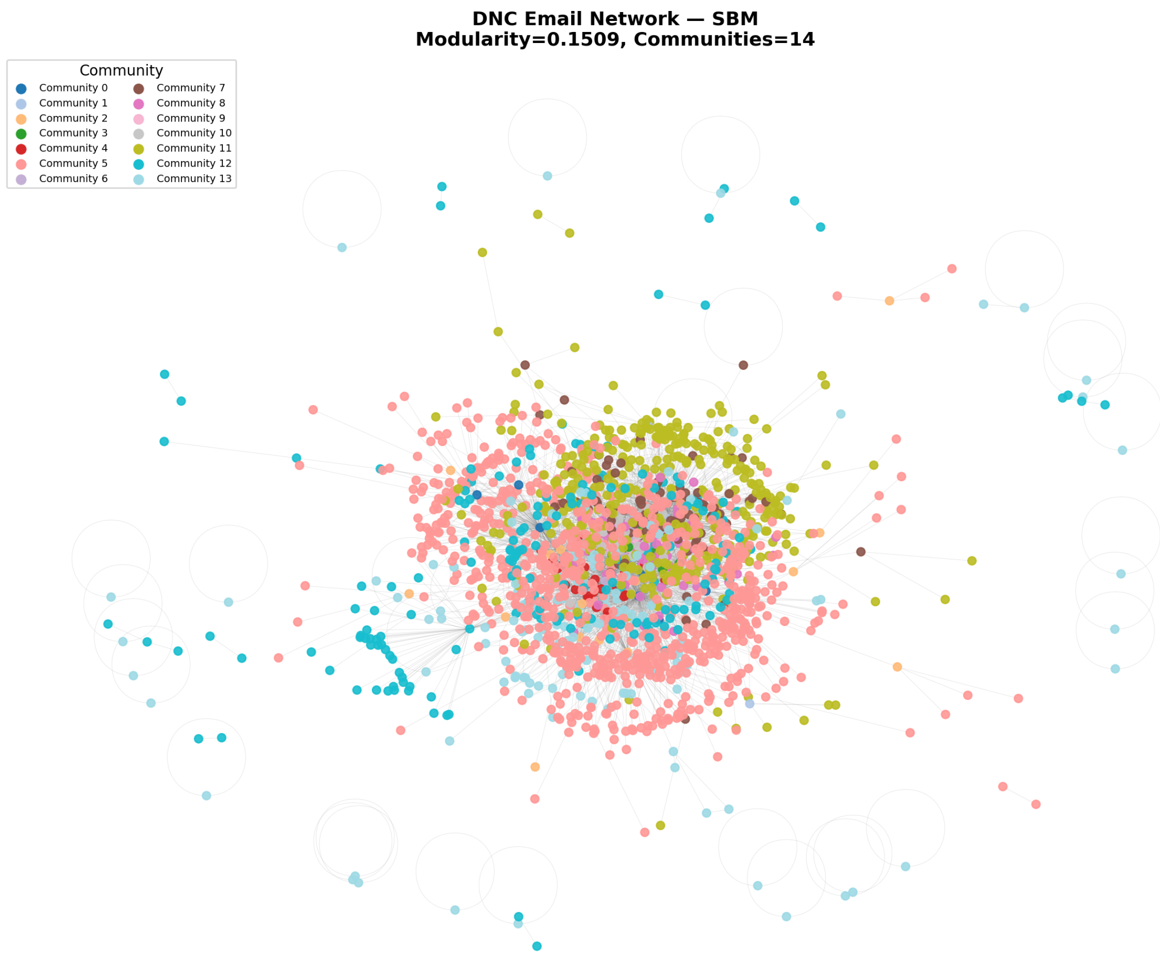
DNC Email Network — SBM
Modularity=0.1509, Communities=14
Community
Community 0
Community 1
Community 2
Community 3
Community 4
Community 5
Community 6
Community 7
Community 8
Community 9
Community 10
Community 11
Community 12
Community 13

Figure 4

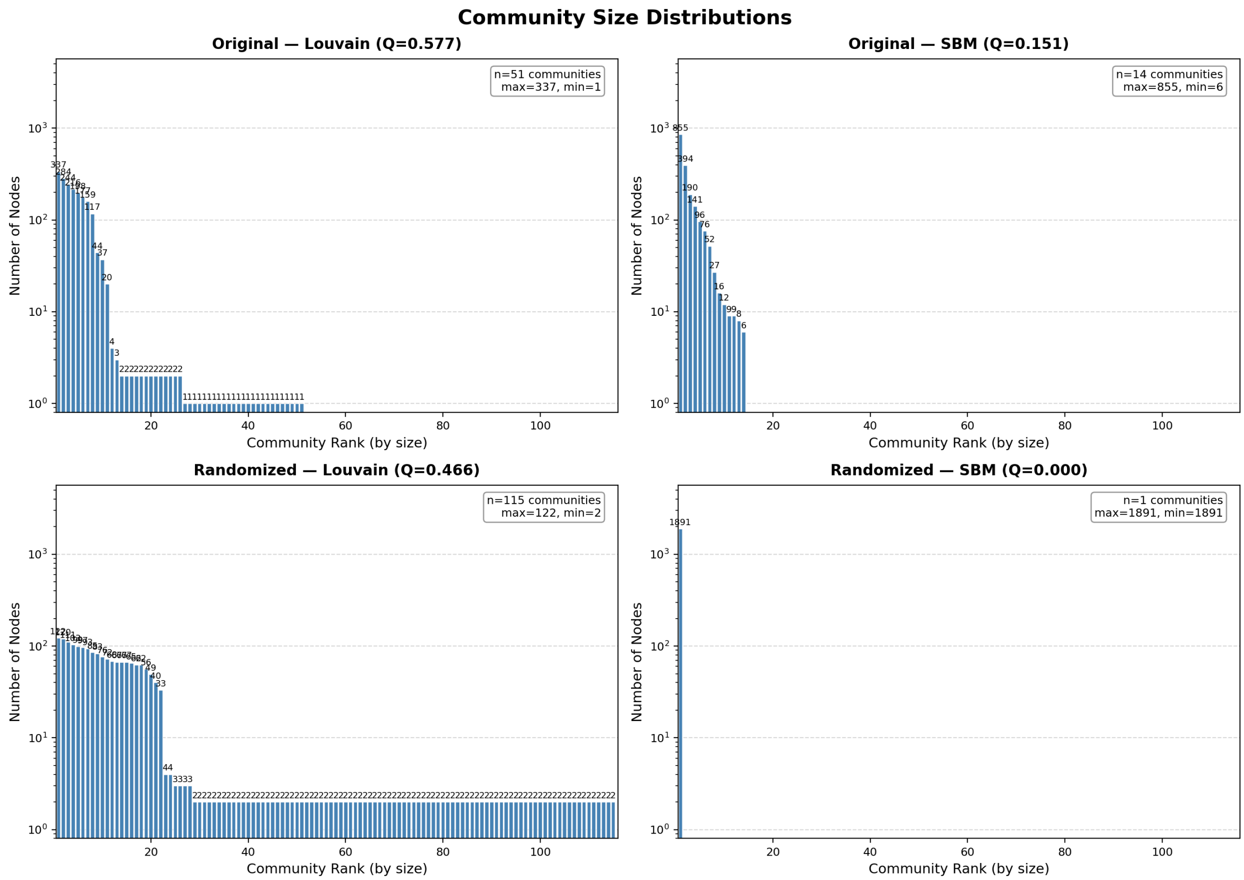

Figure 5

Comparing partitions quantitatively: A good function f(b1, b2) comparing two partitions should have a few features. The function should have a max value when the partitions are identical. The function should have its minimum value when the partitions come from random chance. It is preferable that the function gets normalized so the total range of possible outputs is between zero and 1. It should also "be symmetric" meaning f(b1, b2) = f(b2, b1). The standard approach for comparing community detection results is the Normalized Mutual Information (NMI), (Danon et al. (2005)). NMI handles partitions with different numbers of communities better than simpler measures such as the Rand Index. NMI treats each partition as a random variable and measures how much information one partition tells you about the other. NMI ranges from 0 (completely unrelated partitions) to 1 (identical partitions). If two partitions are randomly assigned, NMI approaches 0. One important caveat is that standard NMI has a bias when partitions have very different numbers of communities. For this reason the literature often uses Adjusted Mutual Information (AMI), which corrects for chance.

4 Community detection and the consensus partition

Many community detection algorithms are stochastic. Consensus clustering addresses this by aggregating many runs into a single stable partition. If two nodes are consistently placed together across runs, that co-assignment is likely meaningful; if only occasionally, their relationship is ambiguous.The consensus matrix C neatly encodes this. For parameters of N nodes and R runs, C of ij catalogs the fraction of runs in which nodes i and j were assigned to the same community. C of ij ranges from 0 (never co-assigned) to 1 (always co-assigned). A threshold is then applied to the matrix C and passed back into a community detection algorithm to produce a final stable partition. This approach was written about in Lancichinetti and Fortunato (2012) in "Consensus clustering in complex networks" (Scientific Reports, 2, 336), who showed it consistently outperforms single runs, especially on networks with weak community structure.

Planted-partition model. Within-block connection probability 0.08. Between-block connection probability 0.005.

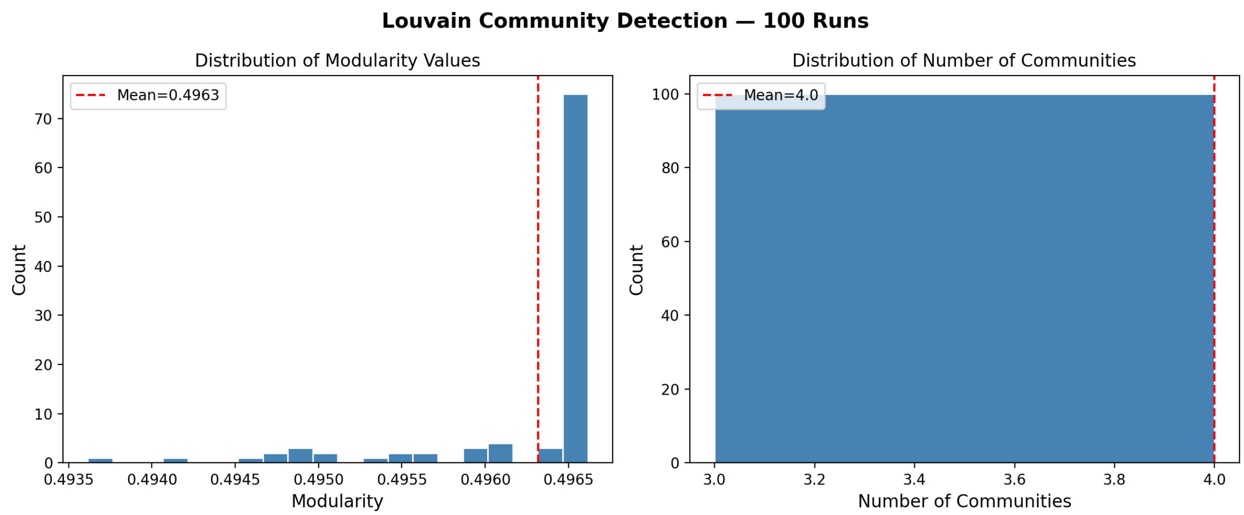

Figure 6

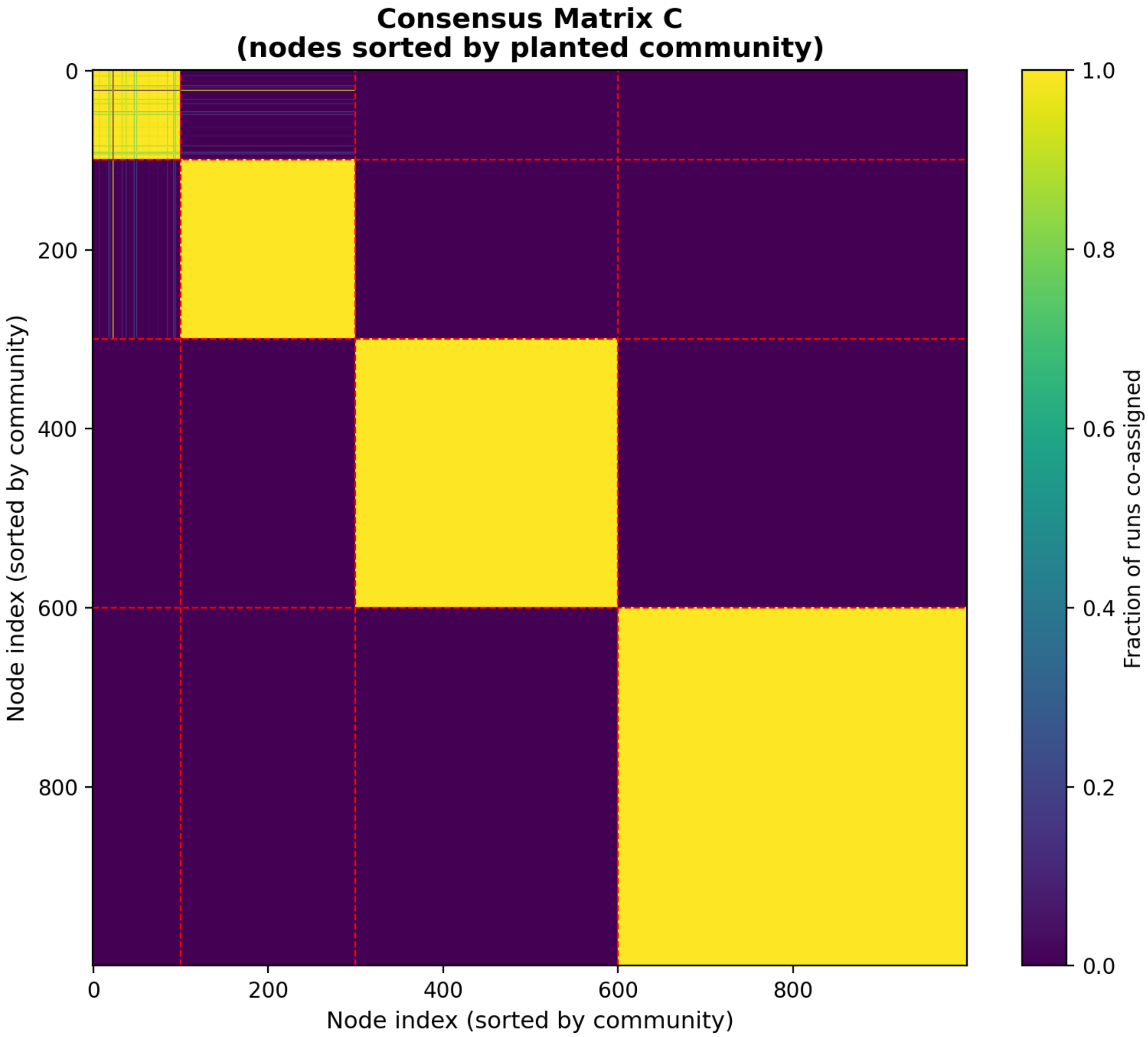
Consensus Matrix C
(nodes sorted by planted community)
0
200
400
600
800
Node index (sorted by community)
Node index (sorted by community)
1.0
0.8
0.6
0.4
0.2
0.0
Fraction of runs co-assigned

Figure 7

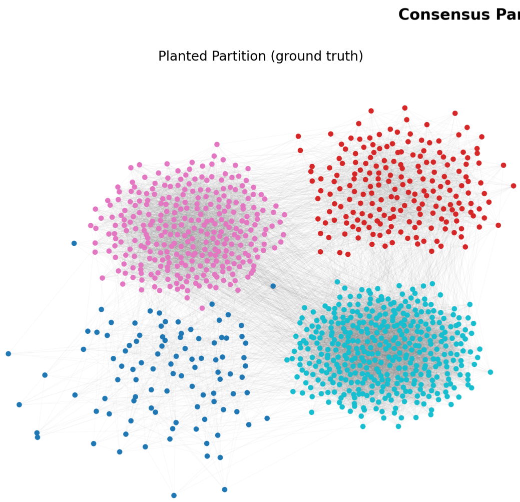

Figure 8: A threshold of 0.5 was chosen with trial and error. I found this value cleanly separated the two groups, keeping only edges where nodes were co-assigned in the majority of runs

Less strong-community structure. Within-block connection probability 0.04. Between-block connection probability 0.002.

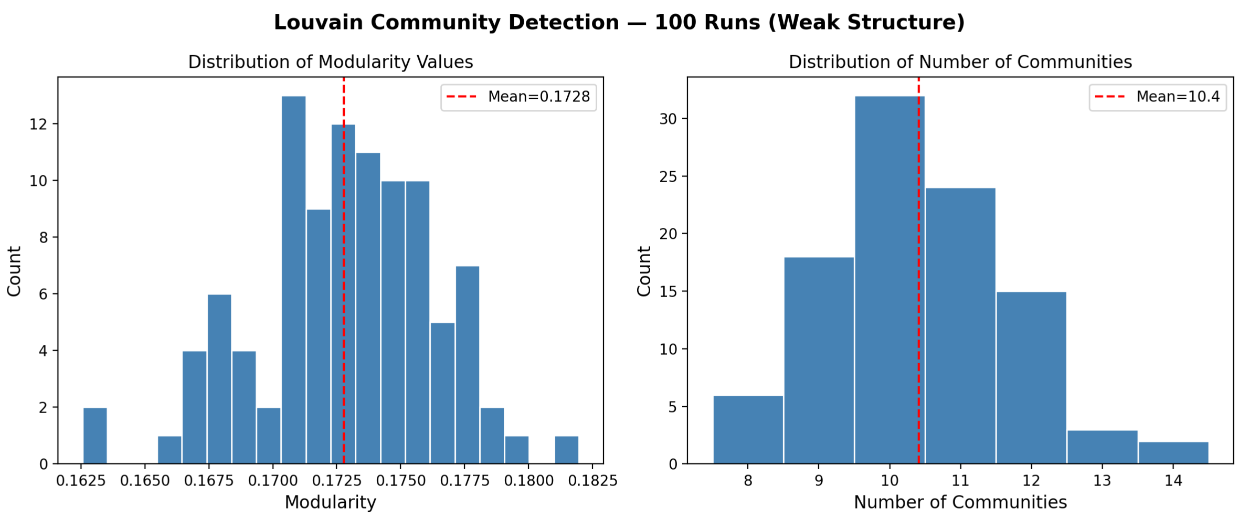

Figure 9

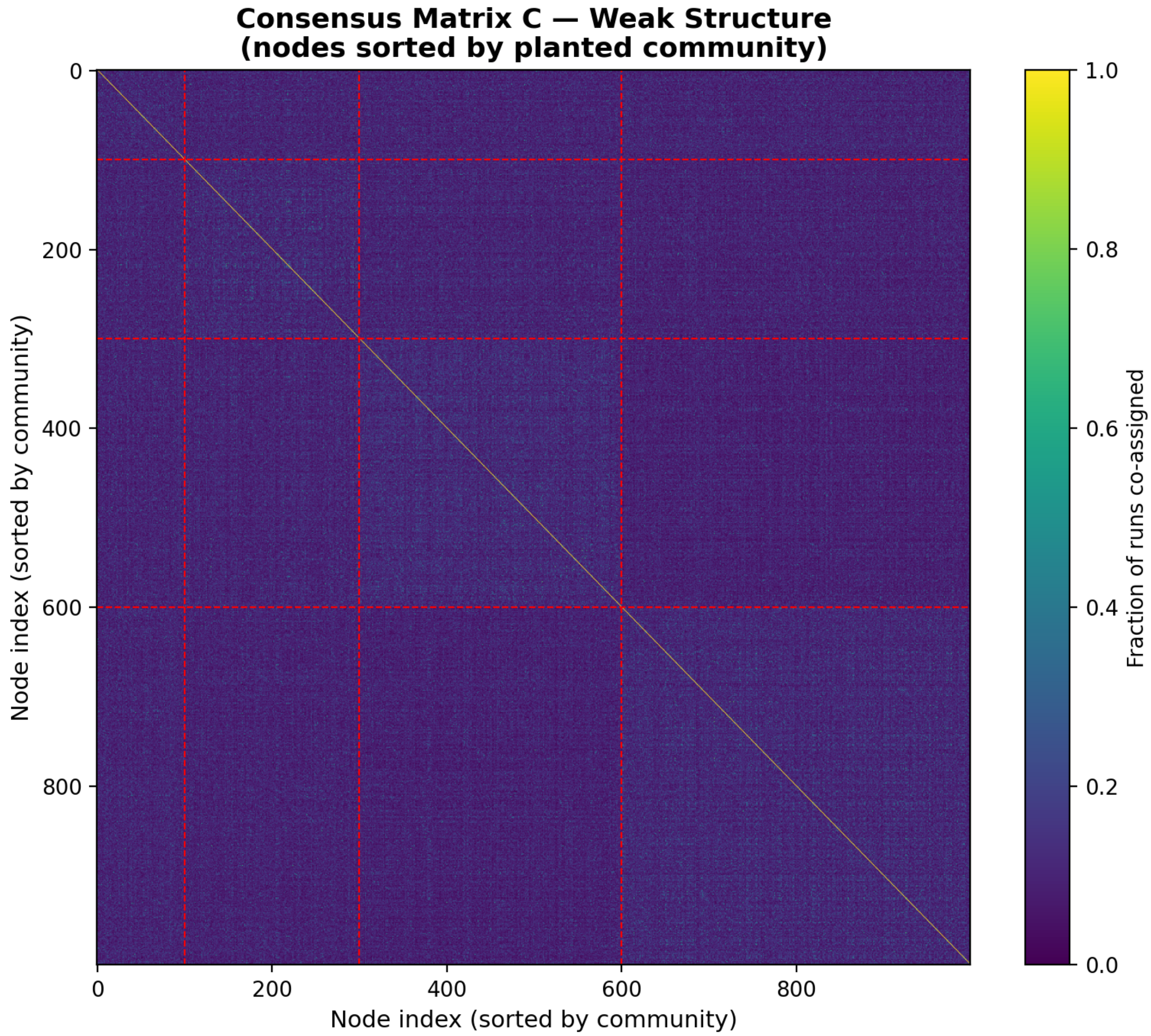
Consensus Matrix C — Weak Structure
(nodes sorted by planted community)
0
200
400
600
800
Node index (sorted by community)
Node index (sorted by community)
0.0
0.2
0.4
0.6
0.8
1.0
Fraction of runs co-assigned

Figure 10

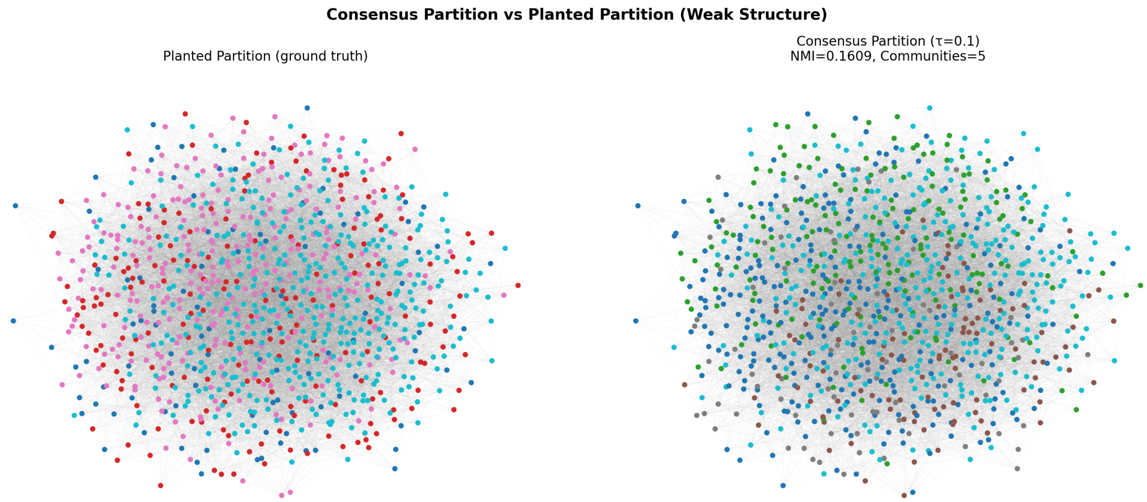

Figure 11: For this case with much weaker structure tau of 0.5 was too strong. Meaning almost no pairs of nodes exceeded 0.5. With trial and error I lowered tau to 0.1 to capture the weaker signal that existed. The meaning is this model retains edges where nodes were co-assigned at least 10 percent of the time.

Does the consensus partition help improve the community detection process? Yes, but the degree of improvement depends on the strength of the underlying community structure. In the strong structure case (p in=0.08, p out=0.005), Louvain was already quite consistent between runs. The consensus partition improved NMI from 0.9695 (single run) to 0.9953, tracking the planted partition exceptionally well. The gain in repeated runs is modest though because a single run was already performing well. The consensus matrix showed clean block-diagonal structure, confirming that aggregating runs added only marginal benefit when the signal is strong and the algorithm is consistent. In the weak structure case (p in=0.04, p out=0.02), the efficacy changed dramatically between the methods. Individual runs of louvain were highly inconsistent. NMI found of only 0.034 against the planted partition. The consensus partition improved NMI to 0.161, which is a large relative improvement over a single run but still low in absolute terms. The consensus matrix showed no clear block structure. Values were uniformly low (around 0.11) with no visible diagonal blocks. This means the 100 runs disagreed so much that aggregation could not recover a clean signal. The key takeaway is that consensus partitioning is most useful when community structure exists but is moderately only ambiguous. It needs to be strong enough to produce some consistent signal across runs, but weak enough that individual runs are unreliable.