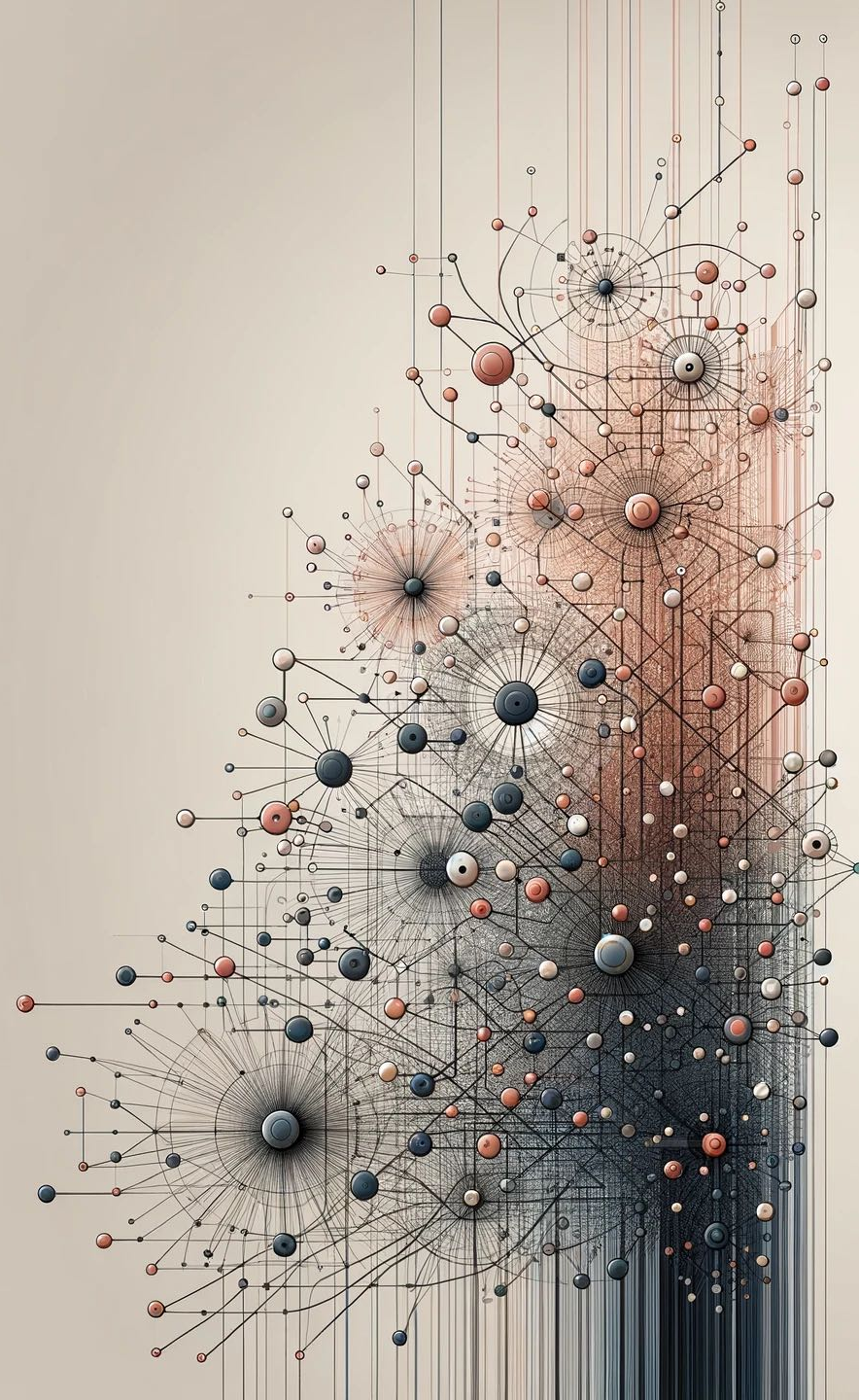# Analisi e Visualizzazione delle Reti Complesse

NS14 - Communities - Benchmarks and Tutorial

Prof. Rossano Schifanella

Outline

- **Method evaluation**

- **Artificial benchmarks**

- **Real benchmarks**

- **Partition similarity**

- **Exercises**

Method evaluation

- **Problem:** How can we quantitatively assess the performance and reliability of a community detection algorithm?

- **Importance:** Evaluation is crucial for comparing algorithms, understanding their strengths/weaknesses, and choosing the appropriate method for a given task.

- **Approach:** Evaluate algorithms on benchmark graphs with known or expected community structures. Performance is measured by comparing the detected communities against the ground truth.

- **Benchmark Types:**
  - **Artificial benchmarks:** Generated networks with controlled community structures. Allow systematic testing under varying conditions.
  - **Real benchmarks:** Empirical networks where community membership is inferred from metadata or external knowledge. Test performance on realistic structures.

Planted partition model

- Stochastic block models (SBMs) provide a general framework for generating graphs with block structures. Often used for creating artificial benchmarks.

```
# Example: network with communities of sizes S
# and connection probabilities defined by matrix P
G = nx.generators.stochastic_block_model(S, P)
```

- The **planted partition model (PPM)** is a specific, widely used SBM variant for community detection benchmarks.

- **Core Idea:** Nodes are pre-assigned to q communities. Edges are placed randomly with probabilities depending *only* on whether the nodes belong to the same community or different communities.
  - p_{int}: Probability of connecting two nodes *within* the same community.
  - p_{ext}: Probability of connecting two nodes in *different* communities.

- **Assumption:** For well-defined communities, we expect $p_{int} > p_{ext}$. The difficulty of detection increases as p_{ext} approaches p_{int}.

Planted partition model: Parameters

- Consider q groups of equal size N/q.

- Expected internal degree (within-community):

$$\langle k^{int} \rangle = p_{int} \left(\frac{N}{q} - 1 \right)$$

- Expected external degree (between-communities):

$$\langle k^{ext} \rangle = p_{ext}(N - \frac{N}{q}) = p_{ext}\frac{N}{q}(q-1)$$

- Expected total degree:

$$\langle k \rangle = \langle k^{int} \rangle + \langle k^{ext} \rangle$$

Planted partition model: Parameters (2)

- **Mixing parameter (μ):** Often used to characterize the community structure's ambiguity. It represents the expected fraction of a node's edges connecting to *other* communities:

$$\mu = \frac{\langle k^{ext} \rangle}{\langle k \rangle} = \frac{\langle k^{ext} \rangle}{\langle k^{int} \rangle + \langle k^{ext} \rangle}$$

  - $\mu \to 0$: Strong, well-separated communities.
  - $\mu \to 1$: Weak or non-existent communities. Detection becomes harder as μ increases.

In NetworkX:

```
# network with q communities of nc nodes each
# and link probabilities p_int and p_ext
G = nx.generators.planted_partition_graph(q, nc, p_int, p_ext)
```

GN benchmark

- A specific instance of the planted partition model proposed by Girvan and Newman (2002).
- **Standard Parameters:**
    - $N = 128$ nodes
    - $q = 4$ communities (each with $N/q = 32$ nodes)
    - $\langle k \rangle = 16$ (average degree fixed)
- Since $\langle k \rangle = \langle k^{int} \rangle + \langle k^{ext} \rangle = p_{int}(31) + p_{ext}(96) = 16$, p_{int} and p_{ext} are coupled. The benchmark varies the relative strength by tuning the mixing parameter $\mu = \langle k^{ext} \rangle / 16$.
- **Construction:** Similar to Erdős–Rényi, edges are placed randomly based on p_{int} and p_{ext}.
- **Historical Significance:** One of the first widely adopted benchmarks for community detection.

GN benchmark: Detectability

- **Challenge:** Detection difficulty increases as μ increases (i.e., $\langle k^{ext} \rangle$ increases).

- **Intuition:** Communities should be detectable if nodes have more internal than external links ($\langle k^{int} \rangle > \langle k^{ext} \rangle$, or $\mu < 0.5$). For $\langle k \rangle = 16$, this corresponds to $\langle k^{ext} \rangle < 8$.

- **Detectability Limit:** Empirical studies show algorithms often fail well before $\mu = 0.5$. The practical limit is around $\langle k^{ext} \rangle \approx 7$ ($\mu \approx 0.44$). Below this, random fluctuations in edge placements obscure the planted structure.
  - Reading: Girvan & Newman, PNAS, 2002.

- **Limitations:** Assumes uniform community sizes and Poisson degree distributions, unlike many real networks.
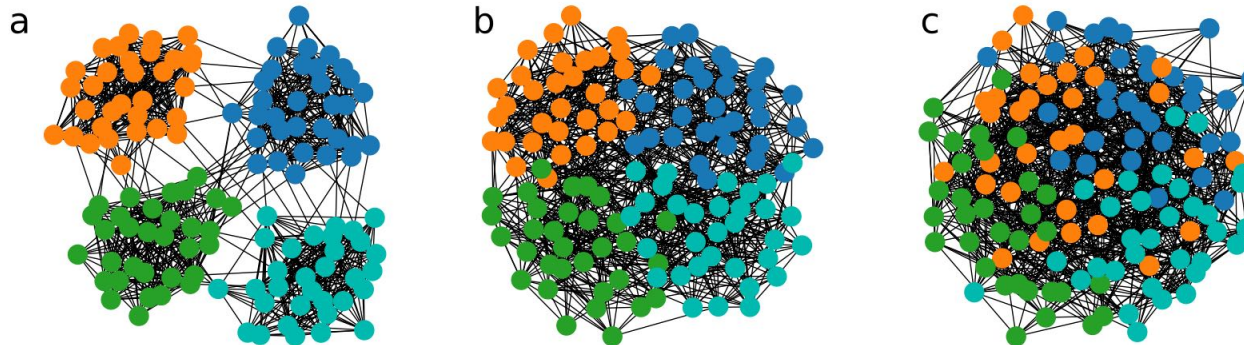
LFR benchmark

- Addresses the limitations of the GN benchmark by incorporating realistic network features.
- **Key Features (Introduced by Lancichinetti, Fortunato, Radicchi, 2008):**
  - **Heterogeneous degree distributions:** Typically power-law distributed $P(k) \sim k^{-\gamma}$.
  - **Heterogeneous community sizes:** Typically power-law distributed $P(s) \sim s^{-\beta}$.
- **Goal:** Generate more realistic benchmarks that better reflect the challenges of community detection in empirical networks.
  - Reading: Lancichinetti et al., PRE, 2008.
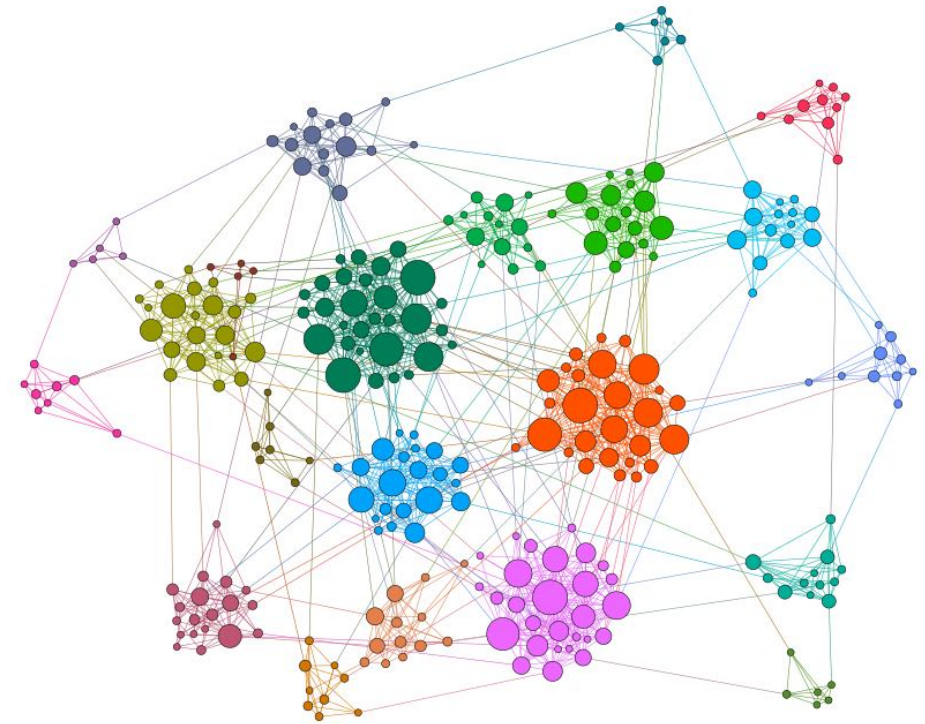
LFR benchmark: Parameters

- Generated using a configuration model approach, respecting specified degree and community size distributions.

- **NetworkX Implementation:**

```
G = nx.generators.LFR_benchmark_graph(
    n, tau1, tau2, mu, average_degree=..., min_degree=..., max_degree=...,
    min_community=..., max_community=..., ...
)
```

- **Key Parameters:**
  - `n` : Number of nodes.
  - `tau1` (γ): Power-law exponent for the degree distribution.
  - `tau2` (β): Power-law exponent for the community size distribution.
  - `mu` (μ): Mixing parameter (average fraction of external degree per node). Controls the difficulty.
  - Other parameters control average/min/max degree and community size.

- **Advantage:** Provides a tunable benchmark capturing key topological properties of real-world networks.

The importance of negative controls

- **Question:** Does the algorithm impose community structure even when none exists?
- **Negative Control:** Test the algorithm on networks known *not* to possess significant community structure (e.g., random graphs).
  - **Common Example:** Erdős-Rényi (ER) random graphs (`nx.erdos_renyi_graph(n, p)`).
- **Desired Outcome:** The algorithm should ideally return a *trivial partition*:
  - One single community containing all nodes.
  - N communities, each containing a single node.
- **Interpretation:** Finding non-trivial communities in random graphs suggests the algorithm is overly sensitive to random fluctuations or may be "overfitting" the network structure. This indicates potential unreliability on real data where structure might be ambiguous.

Real benchmarks: Ground truth communities

- Utilize empirical networks where node groupings ("ground truth" communities) are known from external information or metadata, *not* from the network structure itself.

- **Classic Example: Zachary's Karate Club**
  - Social network of 34 members.
  - Ground truth: The two factions the club split into after a dispute between the instructor (0) and president (33).
  - Task: Can an algorithm recover these factions using only the link structure?

```
G = nx.karate_club_graph()
# Ground truth often stored as node attributes
ground_truth = nx.get_node_attributes(G, 'club')
```
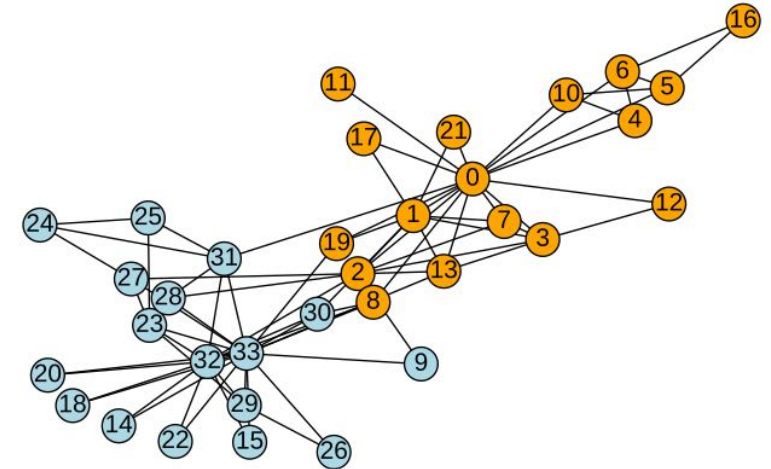
Real benchmarks: Examples & Challenges

- **Other Examples:**
  - **Social Networks:** User-defined groups (e.g., Facebook groups, Twitter lists).
  - **Collaboration Networks:** Co-authors grouped by research field or institution.
  - **Citation Networks:** Papers grouped by journal, conference, or topic.
  - **Biological Networks:** Proteins grouped by functional pathway or cellular location.
  - **Communication Networks:** Email users grouped by department.
- **Core Question:** Do structurally detected communities align with these attribute-based ground truth groups?
- **Challenges:**
  - Ground truth may be incomplete, noisy, or subjective.
  - Structural communities might not perfectly correspond to attribute-based groups (homophily is not always the dominant organizing principle).
  - Evaluating performance requires comparing the algorithm's output partition with the ground truth partition.

Partition similarity measures

- **Problem:** Given two partitions (clusterings) of the same set of N nodes, how similar are they?
    - Partition X: The ground truth communities (from benchmark definition or metadata).
    - Partition Y: The communities found by the detection algorithm.

- **Goal:** Quantify the agreement between X and Y.

- **Desirable Properties:**
    - Value of 1 (or max) for identical partitions.
    - Value of 0 (or min) for unrelated/random partitions.
    - Robustness to the number and size of communities.

- **Common Measures:**
    - Fraction of correctly classified nodes (Simple, but flawed).
    - Normalized Mutual Information (NMI) (Information-theoretic).
    - Adjusted Rand Index (ARI) (Pair-counting based).

Fraction of correctly detected nodes

- The simplest measure of success is just to count the **fraction of nodes that are classified into the correct groups**.

- **Caveat:** group labels can be shuffled

- **Solution:** one commonly calculates the fraction of correctly classified nodes as the maximum over all permutations of the group labels

- It might also be the case that the **number of groups found by the algorithm is not equal** to the number of groups in the ground truth.

    - The same permutation approach works in this case, except that one must **always permute the larger of the two sets of labels**.

Normalized Mutual Information (NMI)

Foundation: Information Theory. Measures the mutual dependence between two partitions.

Intuitive Explanation:

- **H(X)** = "How diverse is partition X?"

- If all nodes are in one community: H(X) is minimal (low diversity)

- If nodes are evenly spread across many communities: H(X) is high (high diversity)

- Think of it as measuring how "surprised" you'd be learning which community a random node belongs to

- **Mathematical definition:** $H(X) = -\sum_{i=1}^{k} p(i) \log p(i)$ where $p(i) = \frac{|X_i|}{N}$ is the probability of a randomly selected node belonging to community i (Shannon's entropy)

- **MI(X,Y)** = "How much knowing Y helps predict X"

- High when communities in X and Y strongly overlap

- Low when knowing Y gives no clue about X

- Example: If X puts students in "grade level" communities and Y puts them in "classroom" communities, MI is high because classrooms contain mostly students from the same grade

- Uses the joint probability distribution of communities in X and Y

Definition:

$$NMI(X,Y) = \frac{2 \times MI(X,Y)}{H(X) + H(Y)}$$

- **Range:** From 0 (completely different) to 1 (identical partitions)

Adjusted Mutual Information (AMI)

- **Problem with NMI:** Standard NMI doesn't account for agreement expected by chance.
- **Solution:** AMI adjusts the mutual information to account for the agreement that would be expected due to random chance alone.
- **Definition:** Similar to ARI, AMI corrects MI by comparing to its expected value under a random model:

$$AMI(X,Y) = \frac{MI(X,Y) - E[MI(X,Y)]}{max(H(X), H(Y)) - E[MI(X,Y)]}$$

where $E[MI(X,Y)]$ is the expected mutual information between random partitions with the same number and sizes of communities as X and Y.

- **Range:**
  - $AMI = 1$: Partitions are identical.
  - $AMI \approx 0$: Agreement is at the level expected by chance.
  - $AMI < 0$: Agreement is less than expected by chance.
- **Advantage:** More robust than NMI against variations in the number and sizes of communities, particularly when comparing partitions with many small communities.

Adjusted Rand Index (ARI)

- **Foundation:** Pair-counting. Compares partitions based on whether pairs of nodes are assigned to the same or different communities.
- **Contingency Table:** Considers all $\binom{N}{2}$ pairs of nodes and classifies them:
    - **a:** Same community in X, Same community in Y.
    - **b:** Same community in X, Different communities in Y.
    - **c:** Different communities in X, Same community in Y.
    - **d:** Different communities in X, Different communities in Y.
- **Rand Index (RI):** Simple agreement fraction: $RI = (a+d)/(a+b+c+d)$.
    - **Problem:** RI approaches 1 even for random partitions as the number of communities increases. It doesn't correct for chance agreement.

Adjusted Rand Index (ARI)

- **Correction for Chance:** ARI adjusts the RI score based on the agreement expected if the partitions were generated randomly (while preserving the number and sizes of communities).

$$ARI = \frac{\text{Index} - \text{Expected Index}}{\text{Max Index} - \text{Expected Index}} = \frac{RI - E[RI]}{1 - E[RI]}$$

- **Range:**
  - $ARI = 1$: Identical partitions.
  - $ARI \approx 0$: Agreement is at the level expected by chance.
  - $ARI < 0$: Agreement is less than expected by chance (possible, but rare).

- **Calculation:** Can be computed efficiently using the contingency table of overlaps (n_{ij}) between communities $i \in X$ and $j \in Y$.

$$ARI = \frac{\sum_{ij} \binom{n_{ij}}{2} - [\sum_i \binom{a_i}{2} \sum_j \binom{b_j}{2}]/\binom{n}{2}}{\frac{1}{2}[\sum_i \binom{a_i}{2} + \sum_j \binom{b_j}{2}] - [\sum_i \binom{a_i}{2} \sum_j \binom{b_j}{2}]/\binom{n}{2}}$$

(a_i: size of community i in X, b_j: size of community j in Y)

Comparing NMI/AMI and ARI

- **Goal:** Quantify similarity between ground truth and detected partitions.
- **NMI/AMI (Normalized/Adjusted Mutual Information):**
  - (+) Information-theoretic basis.
  - (+) Handles partitions with different numbers of communities well.
  - (+) AMI corrects for chance agreement.
  - (-) Can be sensitive to small communities depending on normalization.
- **ARI (Adjusted Rand Index):**
  - (+) Intuitive pair-counting basis.
  - (+) Corrects for chance agreement.
  - (+) Less sensitive to variations in the number of communities than unadjusted RI/NMI.
  - (-) May give less discriminative scores when partitions differ significantly.
- **Recommendation:** Report **both AMI and ARI** for a comprehensive evaluation. They capture different aspects of partition similarity and provide robustness against the limitations of any single metric.

Reading material

[ns1] **Chapter 6 (Communities)**

- References:
  - **Fortunato, S. (2010).** Community detection in graphs. *Physics Reports*, 486(3-5), 75-174. (Comprehensive review)
  - **Girvan, M., & Newman, M. E. J. (2002).** Community structure in social and biological networks. *PNAS*, 99(12), 7821–7826. (Introduced GN benchmark)
  - **Lancichinetti, A., Fortunato, S., & Radicchi, F. (2008).** Benchmark graphs for testing community detection algorithms. *Physical Review E*, 78(4), 046110. (Introduced LFR benchmark)
  - **Danon, L., Díaz-Guilera, A., Duch, J., & Arenas, A. (2005).** Comparing community structure identification. *Journal of Statistical Mechanics: Theory and Experiment*, 2005(09), P09008. (Early comparison study)
  - **Vinh, N. X., Epps, J., & Bailey, J. (2010).** Information theoretic measures for clusterings comparison: Variants, properties, normalization and correction for chance. *Journal of Machine Learning Research*, 11, 2837-2854. (Detailed analysis of NMI/AMI)

Q&A