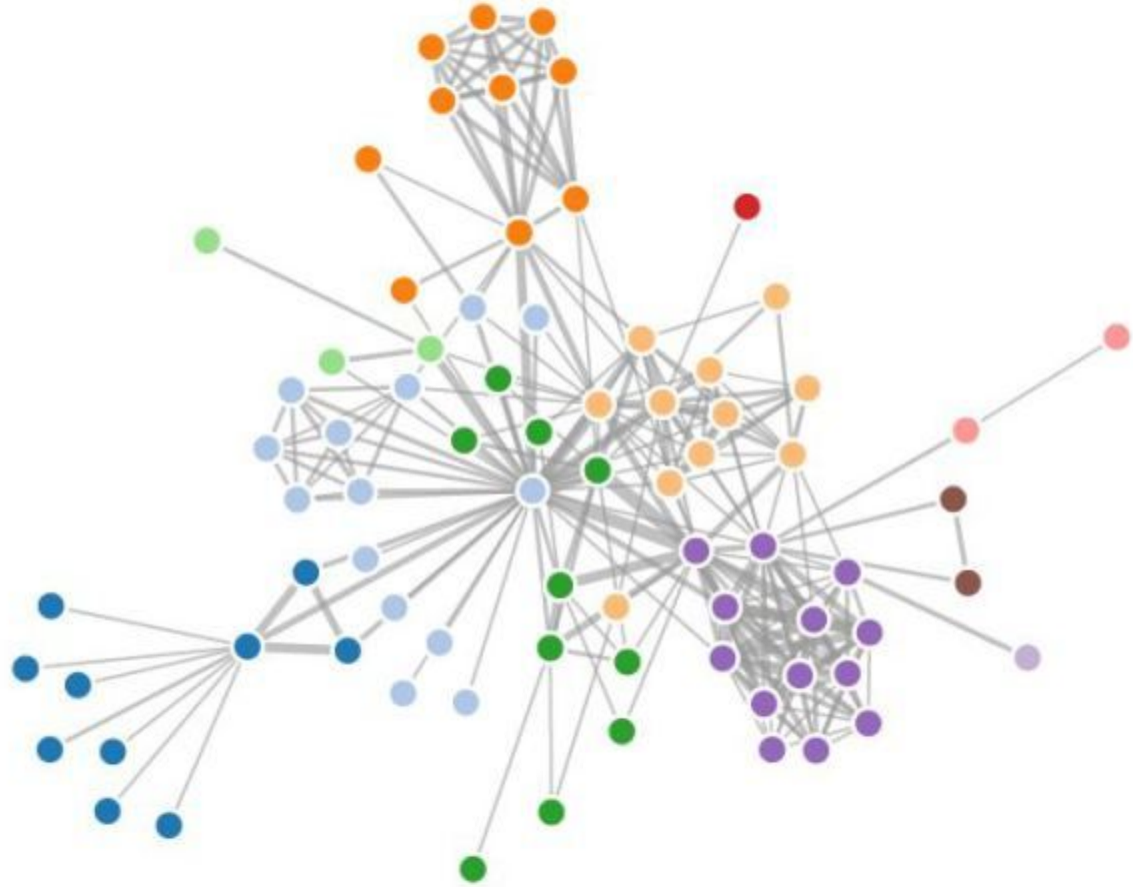
Clustering evaluation



Introduction

Many complex systems of interest such as the Internet, social, and biological relations, can be represented as **networks** consisting a **set of nodes** which are **connected by edges** between them.

Nodes are naturally clustered into **tightly connected modules**, also known as **communities**, with only sparser connections between them.

Many community detection algorithms with different approaches have been proposed in the last decades. Each approach has a **different partitioning strategy** and sometimes different points of view about topological **requirements of a good community**. Consequently, communities detected on a same network by different methods are usually **different in their structural patterns.**

Evaluate clusters using ground-truth

In networks where metadata communities are available, the performance of community detection methods is usually evaluated based on the **similarity** between **communities** that they discover with **metadata communities** considered as **ground-truth**. The most frequently used similarity metrics that are worth mentioning include **Normalized Mutual Information** (NMI), **Recall score**, **Precision score**, **F1-measure**, etc.

However, declared metadata communities are not always good references. Considering them as ground-truth can lead to i**rrelevant conclusions** about the properties of topological community detection methods and **eliminates good structures** that are **not similar** to metadata communities.

- **Precision and Recall**

Given a community set X produced by an algorithm and the ground truth community set Y, for each **community $x \in X$** we **label its nodes** with the **ground truth community $y \in Y$ they belong** to. We then **match community x** with the ground truth community with the **highest number of labels** in the algorithm community. This procedure produces (x, y) pairs having the highest homophily between the node labels in x and all the ground truth communities.

We then measure the quality of the mappings by the two following measures:

- **Precision**: the percentage of nodes in x labeled as y, computed as:

$$P = \frac{|X \cap Y|}{|X|} \in [0,1]$$

- **Recall**: the percentage of nodes in y covered by x, computed as:

$$R = \frac{|x \cap y|}{|y|} \in [0,1]$$

- **F1-measure**

Given a pair (x, y) the two measures describe the **overlap of their members**: a perfect match is obtained when **both precision** and **recall** are **1**. We thus have a many-to one mapping: **multiple communities in X** can be connected to a **single ground truth community in Y**.

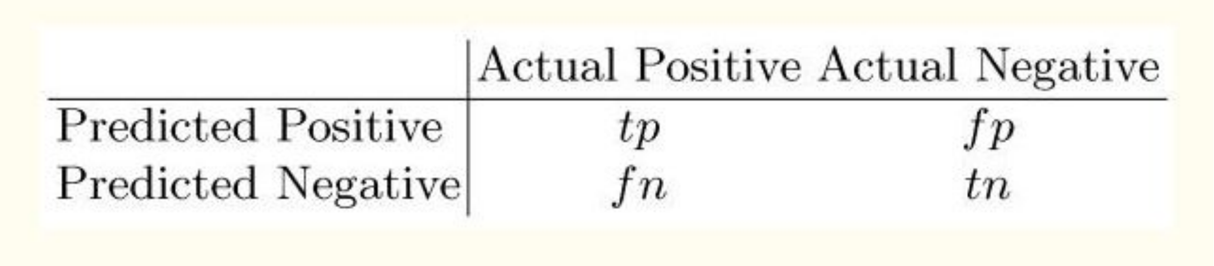
Moreover, analyzing the precision and recall of each pair we are able to detect both **underestimations** and **overestimations** made by the adopted algorithm.

We can combine precision and recall into their harmonic mean obtaining the F1-measure, a concise quality score for the individual pairing:

$$F1 = 2\frac{precision \times recall}{precision + recall}$$

Given a network, the F1 score can be **averaged among all the identified pairs** in order to summarize the overall correspondence between the algorithm community set and ground truth community set.

- **F1-measure**



	Actual Positive	Actual Negative
Predicted Positive	tp	fp
Predicted Negative	fn	tn

$$p = \frac{tp}{tp+fp}$$

$$r = \frac{tp}{tp+fn}$$

$$F1 = \frac{2}{\frac{1}{r}+\frac{1}{p}} = \frac{2tp}{2tp+fp+fn}$$

Evaluate clusters without using ground-truth

Given a set of nodes S, we consider a function $f(S)$ that characterizes how community-like is the connectivity of nodes in S. Let $G(V, E)$ be an undirected graph with $n = |V|$ nodes and $m = |E|$ edges.

Let S be the set of nodes, where n_S is the number of nodes in S, $n_S = |S|$, m_S the number of edges in S, $m_S = |\{(u, v) \in E : u \in S, v \in S\}|$, c_S the number of edges on the boundary of S, $c_S = |\{(u, v) \in E : u \in S, v \notin S\}|$ and $d(u)$ is the degree of node u.

We consider more scoring functions $f(S)$ that capture the notion of quality of a network community S. The experiments we will present later reveal that scoring functions naturally group into the following four classes:

(A) Scoring functions based on internal connectivity:

- **Internal density**:is the internal edge density of the node set S

$$f(S) = \frac{m_s}{\frac{n_S(n_S-1)}{2}}$$

- **Edges inside**: is the number of edges between the members of S

$$f(S) = m_s$$

- **Average degree**: is the average internal degree of the members of S

$$f(S) = \frac{2m_s}{n_s}$$

- **Fraction over median degree (FOMD)**: is the fraction of nodes of S that have internal degree higher than dm, where dm is the median value of d(u) in V

$$f(S) = \frac{|\{u \in S \mid |\{(u,v) | v \in S\}| > d_m\}|}{n_S}$$

- **Triangle Participation Ratio (TPR)**: is the fraction of nodes in S that belong to a triad

$$f(S) = \frac{|\{u \in S \mid \{(v,w) | v,w \in S \ and (u,v),(u,w),(v,w) \in E\} \neq \emptyset\}|}{n_S}$$

(B) Scoring functions based on external connectivity:

- **Expansion**: measures the number of edges per node that point outside the cluster

$$f(S) = \frac{c_S}{n_S}$$

- **Cut Ratio:** is the fraction of existing edges (out of all possible edges) leaving the cluster

$$f(S) = \frac{c_S}{n_S(n - n_S)}$$

(C) Scoring functions that combine internal and external connectivity:

- **Conductance**: measures the fraction of total edge volume that points outside the cluster

$$f(S) = \frac{c_S}{2m_S + c_S}$$

- **Normalized cut:**

$$f(S) = \frac{c_S}{2m_S + c_S} + \frac{c_S}{2(m - m_S) + c_S}$$

- **Flake-ODF: (Out Degree Fraction)**: is the fraction of nodes in S that have fewer edges pointing inside than to the outside of the cluster

$$f(S) = \frac{|\{u \in S \mid |\{(u,v) \in E | v \in S\}| < \frac{d(u)}{2}\}|}{n_S}$$

- **Average-ODF**: is the average fraction of edges of nodes in S that point out of S

$$f(S) = \frac{1}{n_S} \sum_{u \in S} \frac{|\{(u,v) \in E | v \notin S\}|}{d(u)}$$

- **Maximum-ODF (Out Degree Fraction):** is the maximum fraction of edges of a node in S that point outside S

$$f(S) = \max_{u \in S} \frac{|\{(u,v) \in E | v \notin S\}|}{d(u)}$$

(D) Scoring function based on a network model

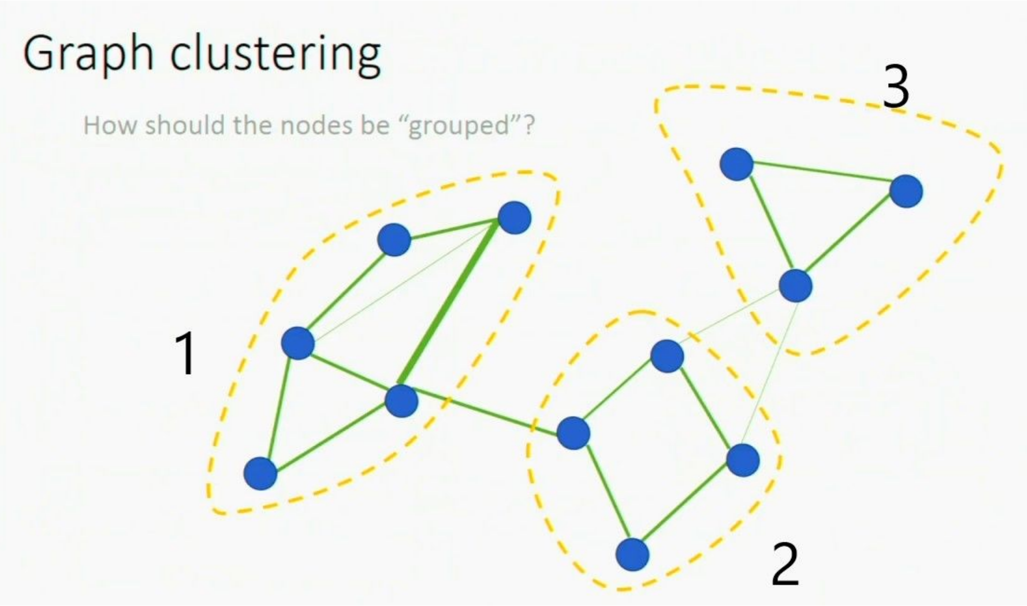
- **Modularity**

To measure the quality of a particular division of a network in k communities using modularity we consider a the matrix **e** whose elements e_{ij} is the fraction of all edges in the network that link vertices in community *i* to vertices in community *j*.

The trace of this matrix $Tr\ e = \sum_i e_{ii}$ is the fraction of the edges in the network that connect vertices in the same community, so clearly a good division into communities should have a high value of this trace.

The trace on its own, however, is not a good indicator of the quality of the division since, for example, placing all vertices in a single community would give the maximal value of $Tr\ e = 1$ while giving no information about community structure at all.

- **Modularity**



Graph clustering

How should the nodes be “grouped”?

For this graph we have the number of vertices equal to 17 and

$$e = \begin{bmatrix} \frac{7}{17} & \frac{1}{17} & 0 \\ \frac{1}{17} & \frac{4}{17} & \frac{2}{17} \\ 0 & \frac{2}{17} & \frac{3}{17} \end{bmatrix}$$

$$Tr(e) = \frac{14}{17}$$

- **Modularity**

So we further define the row (or column) sums $a_i = \sum_j e_{ij}$, which represent the fraction of edges that connect to vertices in community i.

In a network in which edges fall between vertices without regard for the communities they belong to, we would have $e_{ij} = a_i a_j$. This would be the mean case for such a computation. Thus we can define the modularity measure by the formula

$$Q = \Sigma_i (e_{ii} - a_i^2) = \Sigma_i e_{ii} - \Sigma_i a_i^2 = \Sigma_i e_{ii} - \Sigma_i (\Sigma_j e_{ij} \Sigma_k e_{ik}) = Tr(e) - ||e^2||$$

where || **x** || is indicates the sum of the elements of the matrix **x**

- **Modularity**

This quantity measures the fraction of the edges in the network that connect vertices of the same type (i.e., within-community edges) minus the expected value of the same quantity in a network with the same community divisions but random connections between the vertices.

If the number of within-community edges is no better than random, we will get Q = 0. Values approaching Q = 1, which is the maximum, indicate strong community structure. In practice, values for such networks typically fall in the range from about 0.3 to 0.7. Higher values are rare.

- **Modularity**

For the graph we presented above we have

$$||e^2|| = \frac{138}{172}$$

And

$$Q = Tr(e) - ||e^2|| = \frac{14}{17} - \frac{138}{172} = \frac{100}{172} = 0.34$$

- **Modularity density**

Reaserchers Fortunato and Barthélemy pointed out the serious resolution limits of modularity and claimed that the size of a detected module depends on the size of the whole network.

Because of this Zhang proposed another method related to the density of subgraphs. The average modularity is defined as

$$d(G_i) = d_{in}(G_i) - d_{out}(G_i)$$

where $d_{in}(G_i)$ is the average inner degree of the subgraph G_i, which is equal to twice the number of edges in subgraph G_i divided by the number of nodes in set and $d_{out}(G_i)$ is the average outer degree of subgraph G_i , which is equal to the number of edges with one node in the set and the other node outside it divided by the number of nodes in the set

- **Modularity density**

The modularity density formula is defined as the sum of the average modularities for all the subgraphs or partitions in the dataset

$$D = \sum_{k=1}^{n} d(G_i)$$

The best partitioning of the dataset is the one that has the largest D.

For our graph we have

$$d(G_1) = \frac{14}{5} - \frac{1}{5} = \frac{13}{5}$$

$$d(G_2) = 2 - \frac{3}{4} = \frac{5}{4}$$

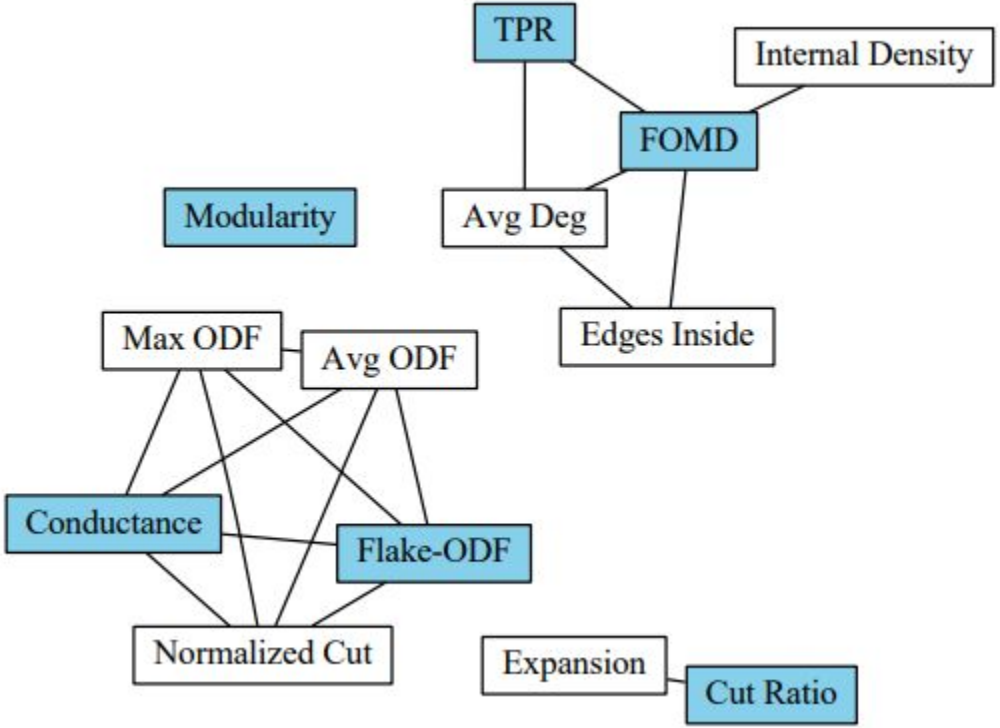
$$d(G_3) = 2 - \frac{2}{3} = \frac{4}{3}$$

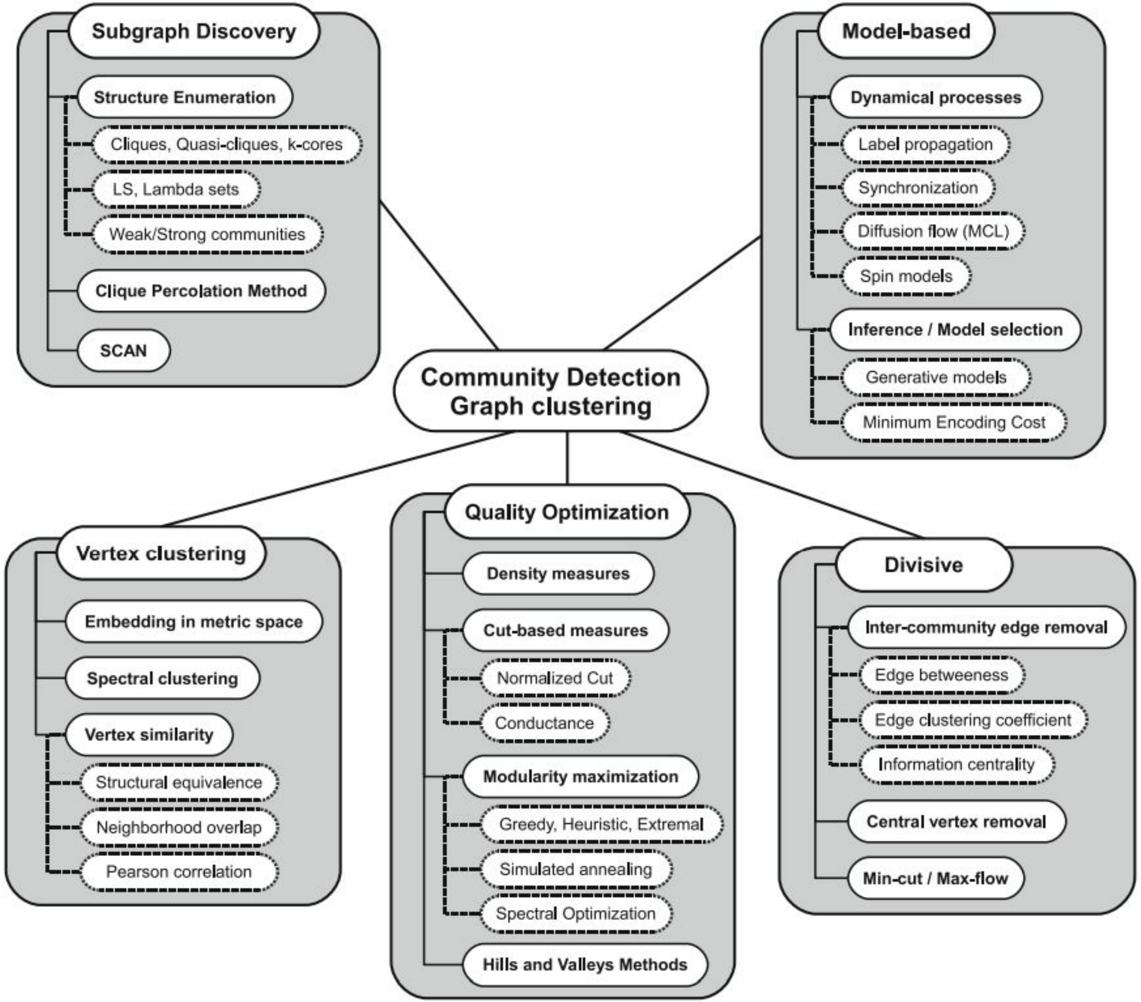
$$D = \sum_{i=1}^{3} d(G_i) = \frac{311}{60} = 5.18$$

Classifying the scoring functions

For about 10 million ground-truth communities, there were computed scores using each of the 13 scoring functions presented.

It could be observed that scores naturally grouped into four clusters. This means that scoring functions of the same cluster return heavily correlated values and quantify the same aspect of connectivity structure. Overall, none of the scoring functions were negatively correlated, which means that none of them systematically disagree. Interestingly, Modularity is not correlated with any other scoring function.


TPR
Internal Density
FOMD
Modularity
Avg Deg
Edges Inside
Max ODF
Avg ODF
Conductance
Flake-ODF
Normalized Cut
Expansion
Cut Ratio


Community Detection
Graph clustering
Subgraph Discovery
Structure Enumeration
Cliques, Quasi-cliques, k-cores
LS, Lambda sets
Weak/Strong communities
Clique Percolation Method
SCAN
Model-based
Dynamical processes
Label propagation
Synchronization
Diffusion flow (MCL)
Spin models
Inference / Model selection
Generative models
Minimum Encoding Cost
Vertex clustering
Embedding in metric space
Spectral clustering
Vertex similarity
Structural equivalence
Neighborhood overlap
Pearson correlation
Quality Optimization
Density measures
Cut-based measures
Normalized Cut
Conductance
Modularity maximization
Greedy, Heuristic, Extremal
Simulated annealing
Spectral Optimization
Hills and Valleys Methods
Divisive
Inter-community edge removal
Edge betweeness
Edge clustering coefficient
Information centrality
Central vertex removal
Min-cut / Max-flow

Sensitivity to changes & noise

There were introduced a couple of community perturbation strategies:

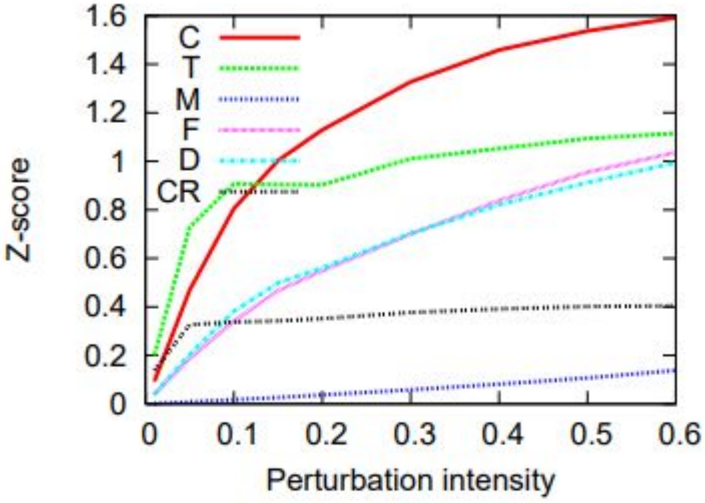
- **NodeSwap**: picking a random edge (u, v), the memberships for u and v were swapped
- **Random**: takes community members and replaces them with random non-members
- **Expand**: perturbation grows the membership set S by expanding it at the boundary
- **Shrink**: removes members from the community boundary

For 6 of the community scoring functions, a score (Z-score) was measured for perturbation intensity **p** ranging between 0.01 and 0.6. This means that we randomly swap between 1% and 60% of the community members, by using each of the perturbation strategies mentioned.

For small p, small Z-scores are desirable since they indicate that the scoring function is **robust to noise**. For high perturbation intensities p, high Z-scores are preferred because this suggests that the community scoring function is **sensitive**, i.e., as the community becomes more “random” we want the scoring function to significantly increase its value.

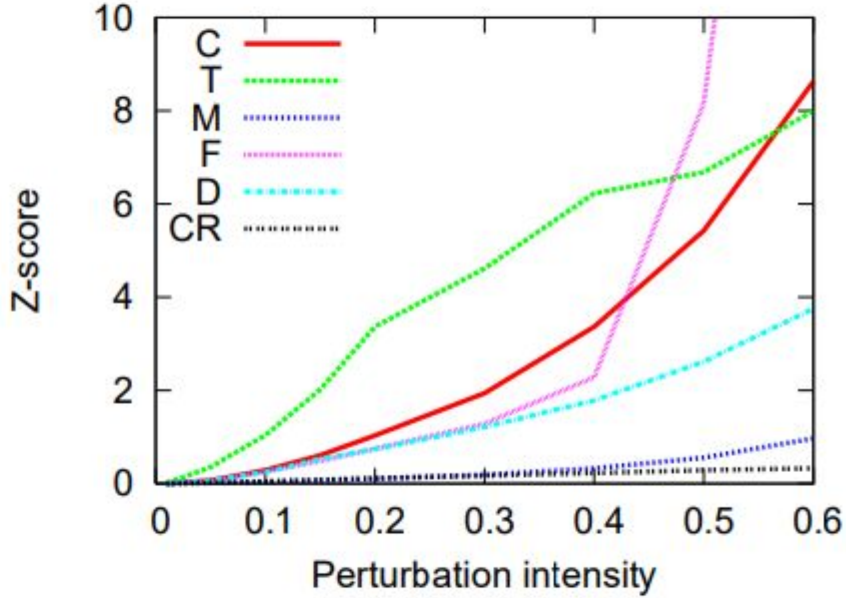
The results were the following:

NodeSwap



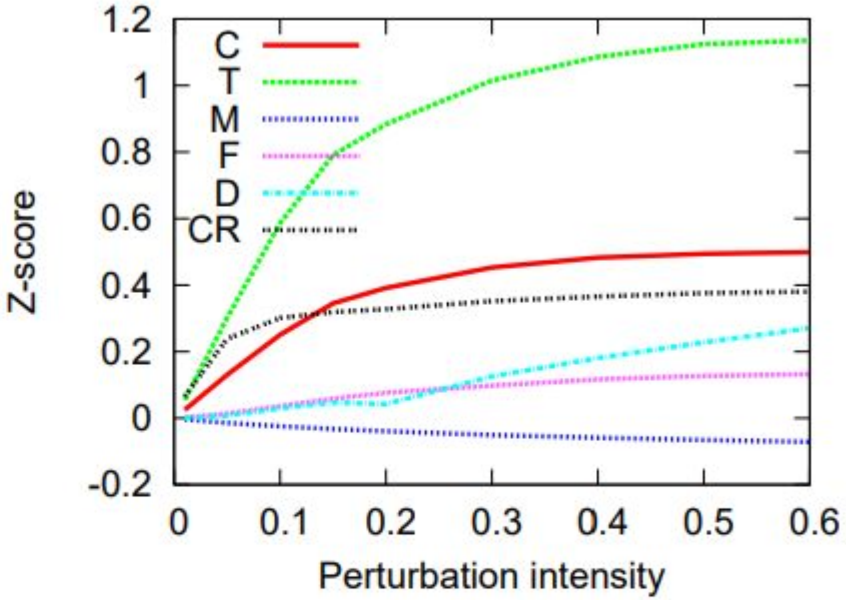
Conductance (C)
Flake-ODF (F)
FOMD (D)
Triad Participation Ratio (T)
Modularity(M)
CutRatio (CR)

Random



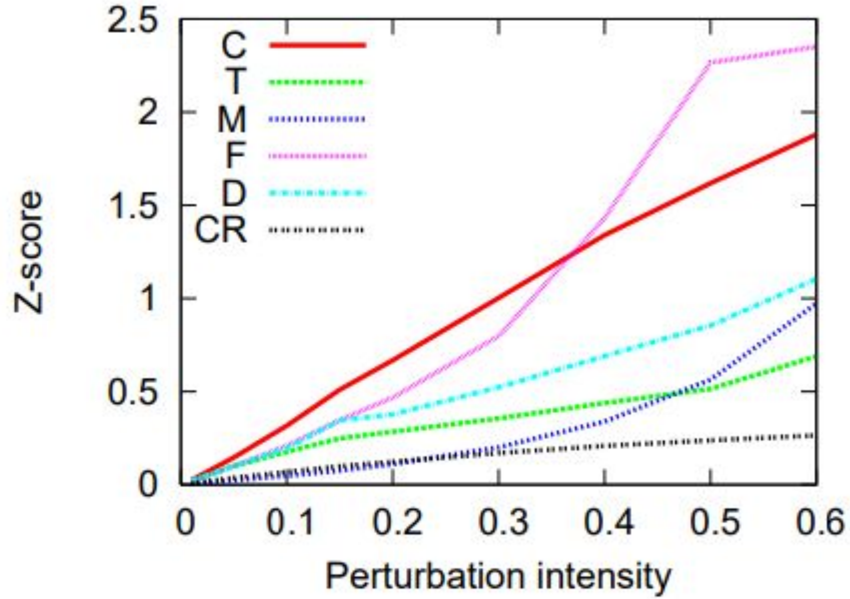
Conductance (C)
Flake-ODF (F)
FOMD (D)
Triad Participation Ratio (T)
Modularity(M)
CutRatio (CR)

Expand



Conductance (C)
Flake-ODF (F)
FOMD (D)
Triad Participation Ratio (T)
Modularity(M)
CutRatio (CR)

Shrink



Conductance (C)
Flake-ODF (F)
FOMD (D)
Triad Participation Ratio (T)
Modularity(M)
CutRatio (CR)

Modularity (M) score does not change much as we perturb the ground-truth communities. This means that Modularity is **not good at distinguishing true communities** from **randomized sets of nodes**. We note very similar results on all of the remaining datasets considered in this study.

Conductance is the **most robust** score under NODESWAP and SHRINK. The **Triad Participation Ratio** (T) is the most robust under RANDOM and EXPAND. In both cases Conductance follows them closely.

Bibliography

- https://arxiv.org/pdf/1205.6233.pdf
- https://hal.archives-ouvertes.fr/hal-01577343/document
- http://www.giuliorossetti.net/about/wp-content/uploads/2015/12/Complenet16.pdf
- https://pdfs.semanticscholar.org/807a/e918cf88325424f08f8cebb4d1007c282b05.pdf
- https://arxiv.org/pdf/cond-mat/0308217.pdf
- https://www.researchgate.net/profile/Shihua_Zhang/publication/5334566_Quantitative_function_for_community_detection/links/00b4953c49ad6d3160000000/Quantitative-function-for-community-detection.pdf
- https://repository.upenn.edu/cgi/viewcontent.cgi?article=1101&context=cis_papers