Comp790-166: Computational Biology

Lecture 4

January 22, 2023

Today

- Finish up modularity
- Finding biologically-relevant modules in data repositories
- Probabilistic graph partitioning approaches- stochastic block model and affiliation model

Do You Remember Question

Last time we met LargeVis, a graph embedding algorithm. What is the intuition for how distance in the embedding relates to the structure of the graph?

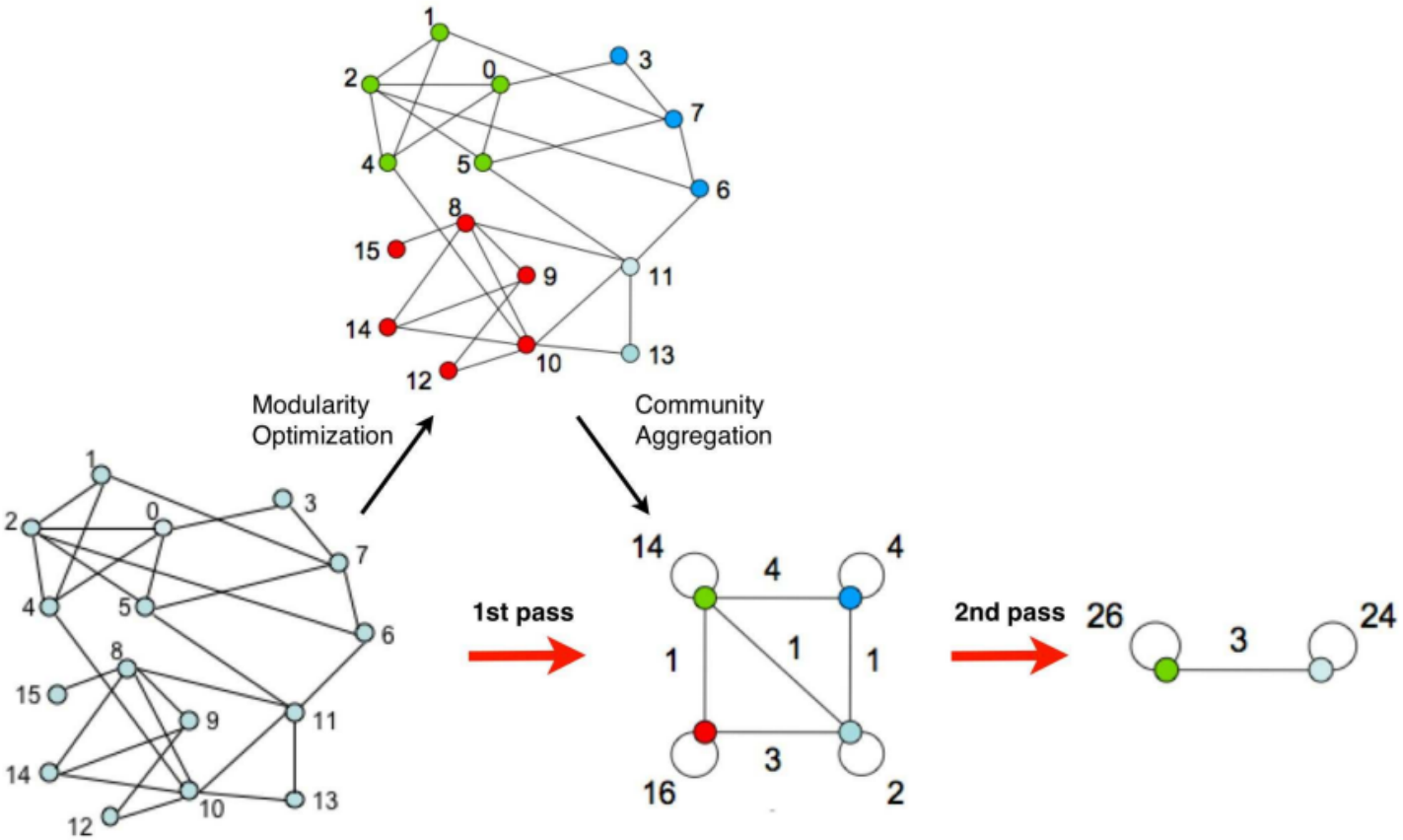
Modularity Defined

$$Q = \frac{1}{2M} \sum_{i,j} [A_{ij} - \gamma \frac{k_i k_j}{2M}] \delta(c_i, c_j) \tag{1}$$

- A_{ij} is the adjacency matrix entry for node pair (i, j) (can be weighted and not just binary)
- $\delta(c_i, c_j)$ is an indicator function for whether or not nodes i and j were assigned to the same community.
- We need an algorithm to help us determine node-to-community assignments for all nodes i, such that Q is as large as possible.
- γ is a resolution parameter controlling the size of communities

Louvain: A Simple Algorithm that Makes a lot of Sense

Merge (if modularity increases), agglomerate, repeat until modularity doesn't increase anymore.



Louvain is Fast Because Potential Merges are Easy to Compute

They show in Blondel *et al. 2008* that the change in modularity by moving a node into a community, C, can be computed in closed form as,

$$\Delta Q = [\frac{\sum_{in} + k_{i,in}}{2M} - (\frac{\sum_{tot} + k_i}{2M})^2] - [\frac{\sum_{in}}{2M} - (\frac{\sum_{tot}}{2M})^2 - (\frac{k_i}{2M})^2] \quad (2)$$

- $\sum_{in}$ the number of edges (or sum of the weights) of links inside of community C
- $\sum_{tot}$ is the number (or sum of the weights) of the edges connected to the nodes in C.
- k_i is the degree of node i
- $k_{i,in}$ is the sum of the edges (or edge weights) from nodes i to nodes in C

Practical Louvain Details

- Very fast, scalable, method. Works for most things if your graph is relatively sparse and or structured.
- Code: `https://pypi.org/project/louvain/`
- A very good bet to get the job done quickly....
- You do not need to specify the number of communities. The default resolution parameter is $\gamma = 1$.
- A limitation is that is only allows for a hard partition of nodes.

All was Fine and Good Until they Realized a little Quirk

Louvain can produce communities that are internally disconnected. That is, the shortest path between a pair of nodes in the same community may require actually leaving the community.

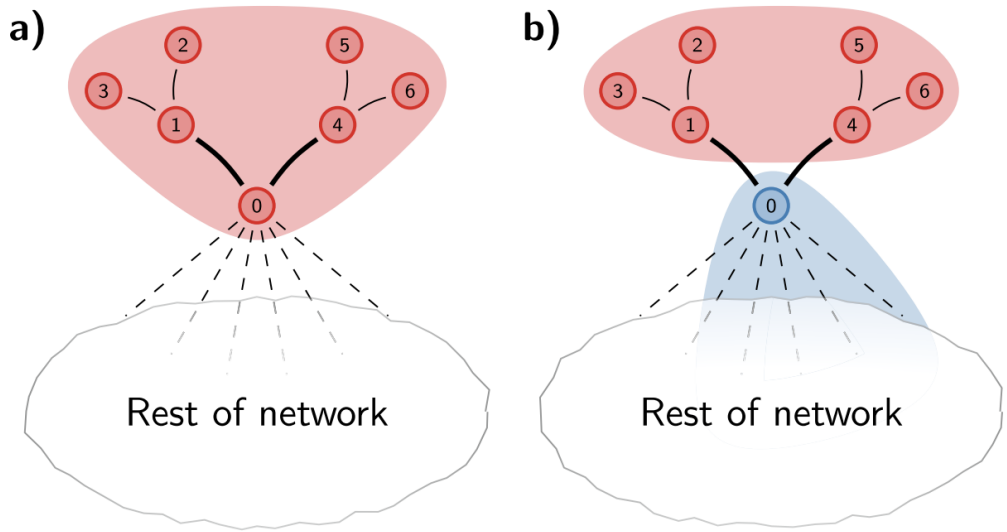


Figure: from Traag *et al.* Scientific Reports. 2018.

Overview of Leiden

Leiden makes a few modifications to guarantee well-connected communities.

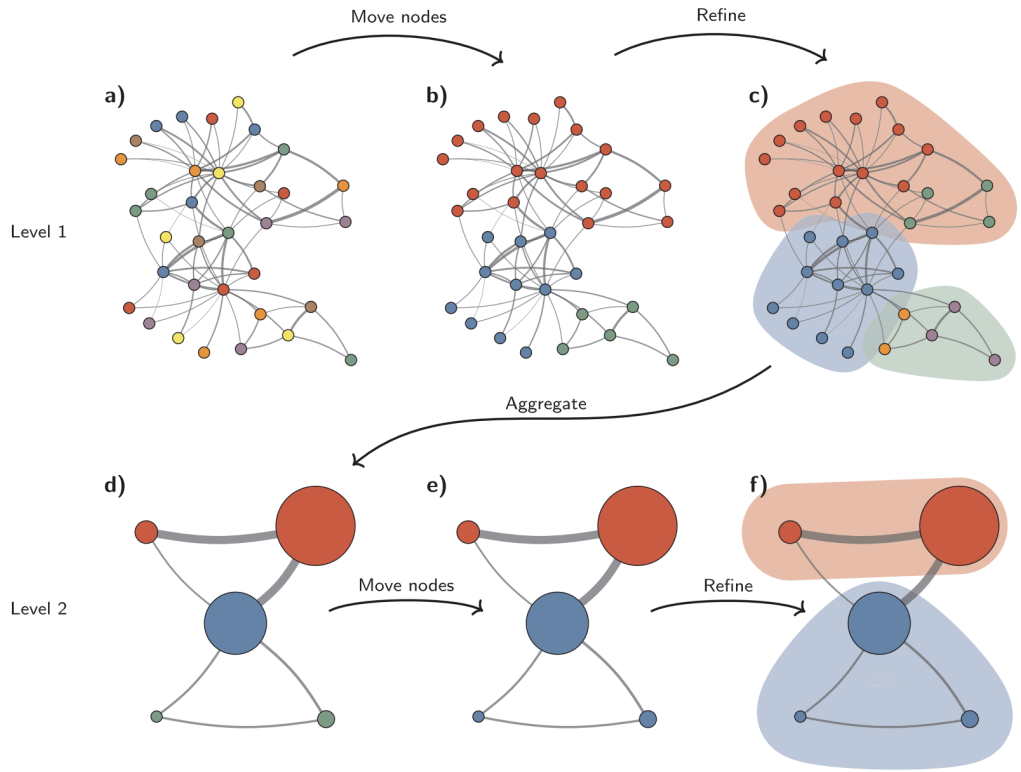


Figure: from Traag *et al.* Scientific Reports. 2018.

Methods of Clean-Up (1)

- **Smart Local Move:** Regular Louvain update of a merge, such that objective function cannot be increased
  - Before aggregating, consider the possibility of splitting each newly formed community
  - Gives the opportunity to split a subset of the newly formed community

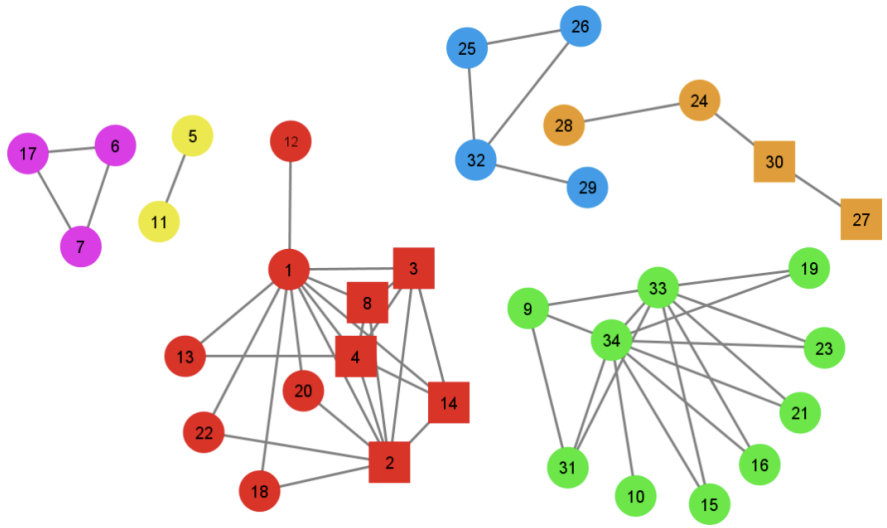


Figure: from `https://arxiv.org/pdf/1308.6604.pdf`

Methods of Clean-Up (2)

- **Random Moving:** For a particular node chosen at random, look at its neighbor in a different community and see if there is a gain in modularity if moving to that community
  - Intuition : The best community to join is the one where most of your neighbors are.

Punchline: Leiden will find higher quality communities (e.g. better connected) in less time than Louvain.
Code: `https://github.com/vtraag/leidenalg`

scanpy.tl.leiden

scanpy.tl.leiden(*adata, resolution=1, *, restrict_to=None, random_state=0, key_added='leiden', adjacency=None, directed=True, use_weights=True, n_iterations=-1, partition_type=None, neighbors_key=None, obsp=None, copy=False, **partition_kwargs*)

Cluster cells into subgroups [Traag18].

Cluster cells using the Leiden algorithm [Traag18], an improved version of the Louvain algorithm [Blondel08]. It has been proposed for single-cell analysis by [Levine15].

This requires having ran `neighbors()` or `bbknn()` first.

A Word of Caution for Single-Cell People

It is incorrect to say 'I clustered by single-cell data with scanpy' or to say 'I clustered by single-cell data with Seurat.' Look at which graph partitioning approach they were using!

Indeed, there are less disconnected communities under leiden...

Depending on your application, this is important. At some point though if your graph gets large enough, what do we think?

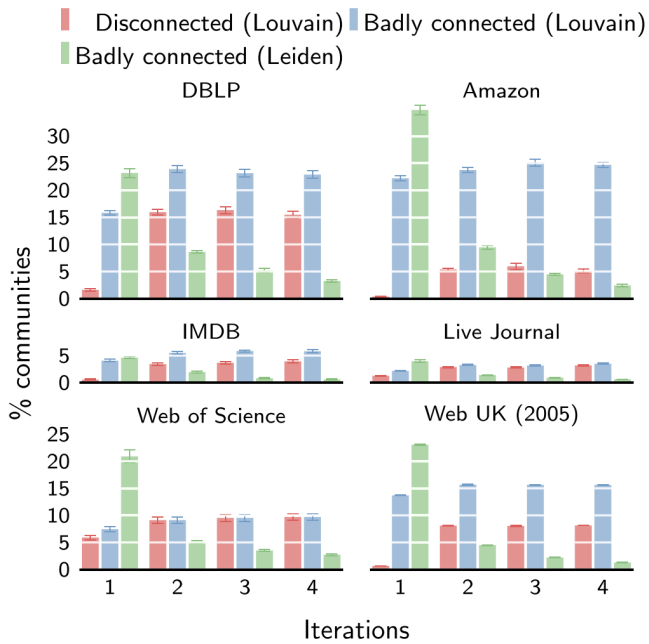


Figure: from Traag *et al.* Scientific Reports. 2018.

Badly connected $\rightarrow$ two nodes in the same community can be connected by walking outside of the community.

A Question

Can anyone think of a situation where disconnected communities might still be valid to consider?

Leiden Also Increases the Maximum Observed Modularity

	Nodes	Degree	Max. modularity	
			Louvain	Leiden
DBLP	317,080	6.6	0.8262	0.8387
Amazon	334,863	5.6	0.9301	0.9341
IMDB	374,511	80.2	0.7062	0.7069
Live Journal	3,997,962	17.4	0.7653	0.7739
Web of Science	9,811,130	21.2	0.7911	0.7951
Web UK	39,252,879	39.8	0.9796	0.9801

Figure: from Traag *et al.* Scientific Reports. 2018.

Modularity is more of a big picture score. I think whether you care about modularity or disconnectedness depends on your application.

Free Project Idea

A free idea for a course project: The authors define here a very particular notion of quality community. Are there other quality definitions. How does this observation change with different kinds of networks?

- You can find many of the standard networks people using for benchmarking and more on SNAP
  `https://snap.stanford.edu/data/`
- You can also use the biological networks in Choobdar *et al.*
  `https://www.nature.com/articles/s41592-019-0509-5`

Partitions of Biological Graphs

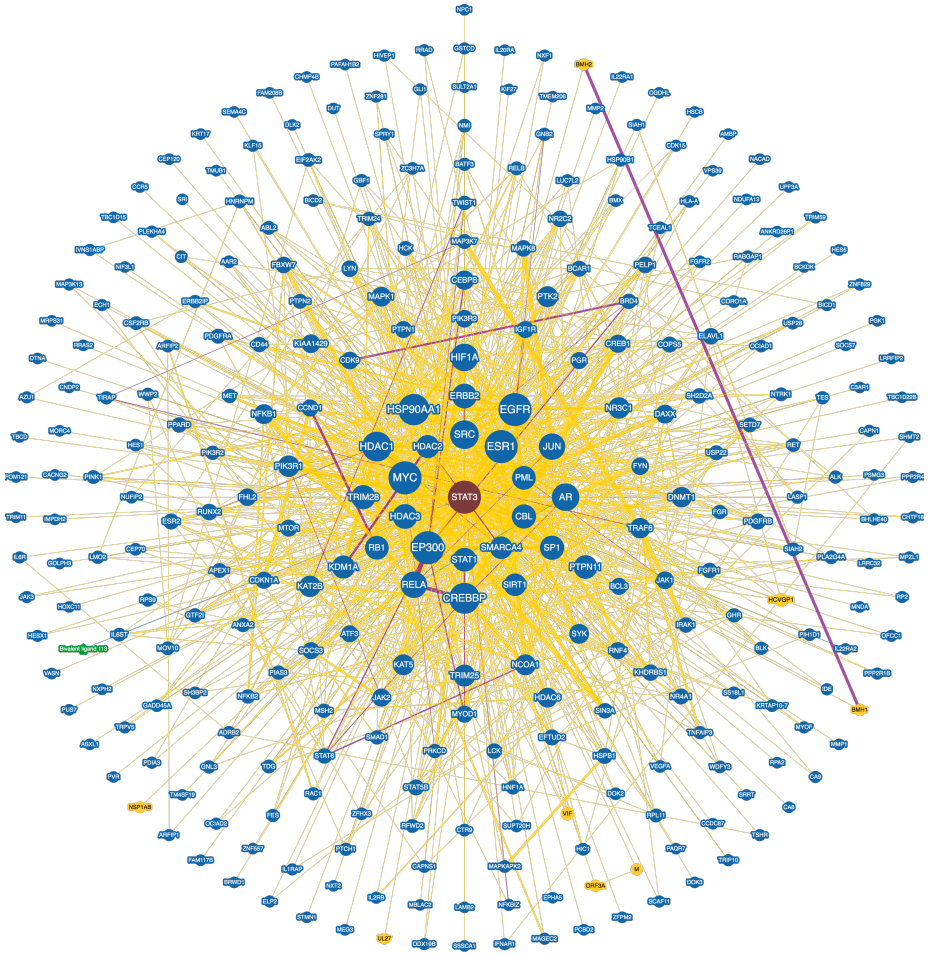


Figure: We are interested in the interactions with STAT3. These can be measured in different ways (gene expression, protein expression, physical interaction)

How Does Modularity-Based Optimization do in the Biological Network Benchmarking Challenge?

Here there is a really nice ‘ground truth‘ understanding, which is gene and protein interactions linked to particular diseases.

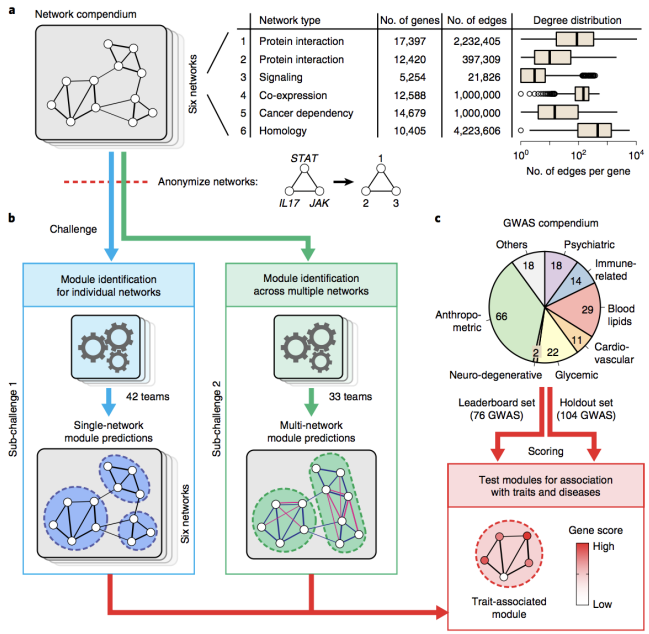
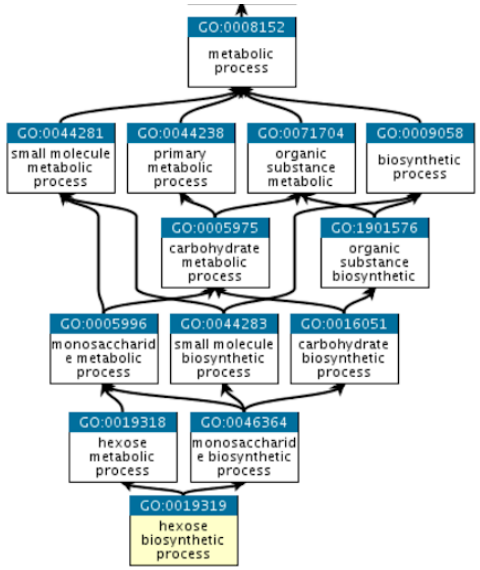


Figure: from Choobar *et al.* Nature Methods 2018.

Good Places to Look for Ground Truth to Validate Your Methods

- Gene Ontology → genes that are involved in the same pathway.
- MSigDB → genes related to certain categories (cancer, immune system, regulatory targets, etc.), text mined
- KEGG → metabolism
- Panther → genes and proteins



It seems that modularity is not only numerically useful

We use modularity to effectively score a candidate partition. This experiment shows that modularity is indeed correlated with the identification of biologically meaningful modules.

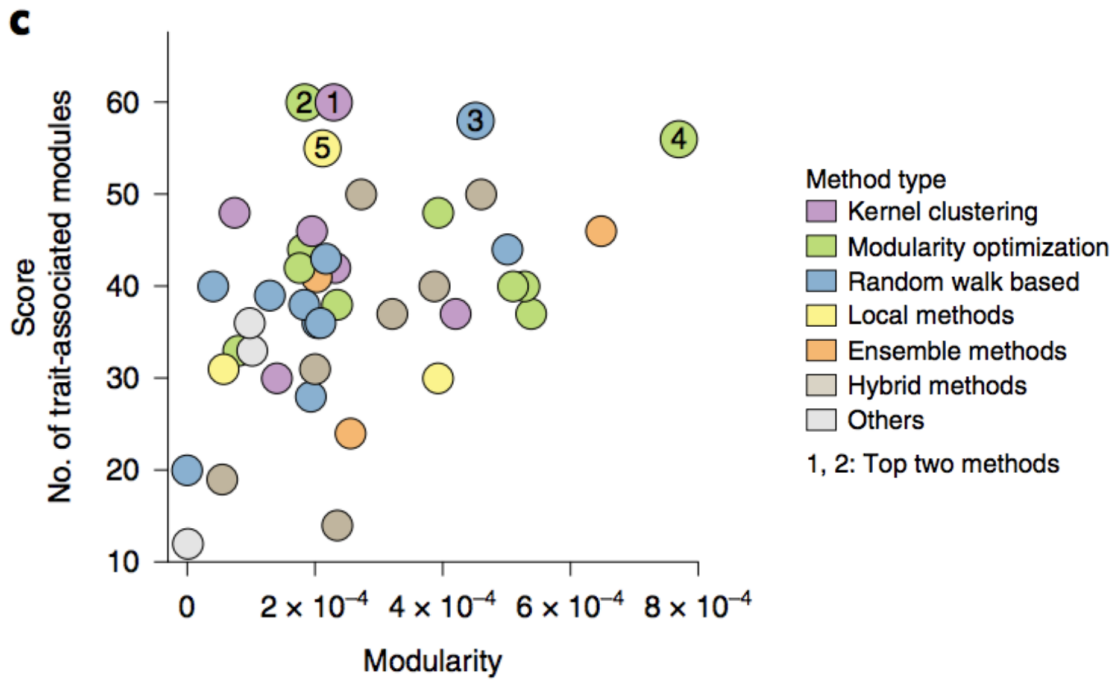


Figure: from Choobar *et al.* Nature Methods 2018.

Examples of Biologically-Relevant Modules

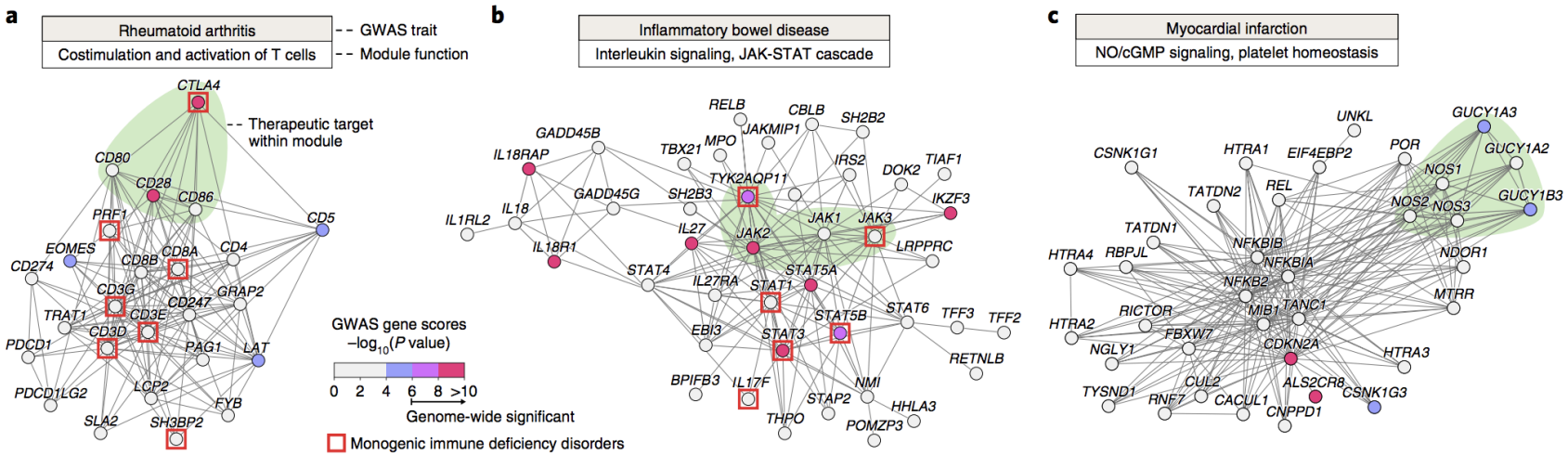


Figure: from Choobar *et al.* Nature Methods 2018.

Transition to two probabilistic optimization approaches where a partition is optimized in a way that can best recapitulate the observed edges.

Edges as Coin Flips

- I want to grab a pair of nodes and guess whether or not there should be an edge between them
- We could use clusters assignments of nodes to make such a decision
  - An OK assumption because we assume that nodes should be more connected within a cluster than between
- We will model the probability that an edge exists between nodes based on the cluster assignments of the nodes
- issue: We at first don't know the cluster assignments for nodes!

Stochastic Block Model (SBM)

- **Intuition:** Members of a community should be connected to themselves and to members of other communities in the same way.
- **Model:** Assuming we have N nodes and K communities, we infer two main parameters
  - $\boldsymbol{\theta} \in \mathbb{R}^{K \times K}$, a matrix of between-community edge probabilities
  - $\mathbf{z} \in \mathbb{R}^{N \times 1}$, a vector of node-to-community assignments

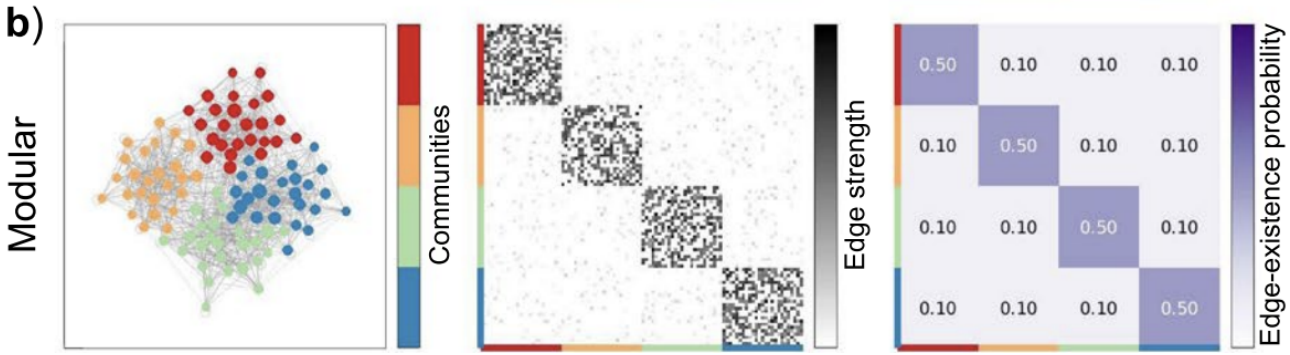


Figure: Notice the high within-edge probability on the diagonal of $\boldsymbol{\theta}$. from Fastkowitz *et al.* Scientific Reports. 2018.

Let's Try to Guess Some θs

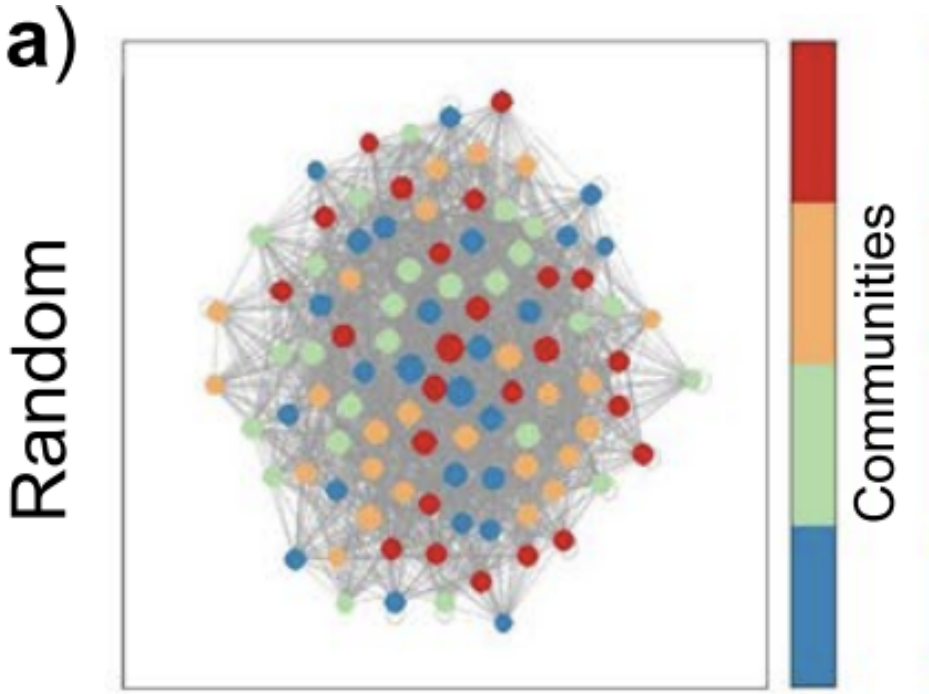


Figure: No structure (this is in fact a random graph). Connections between all kinds of nodes.

A Harder θ

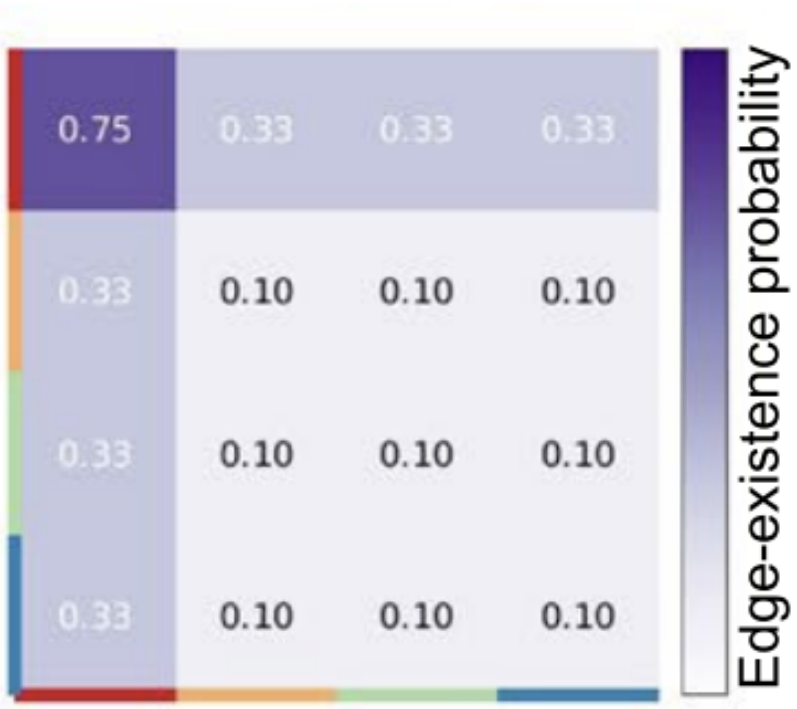
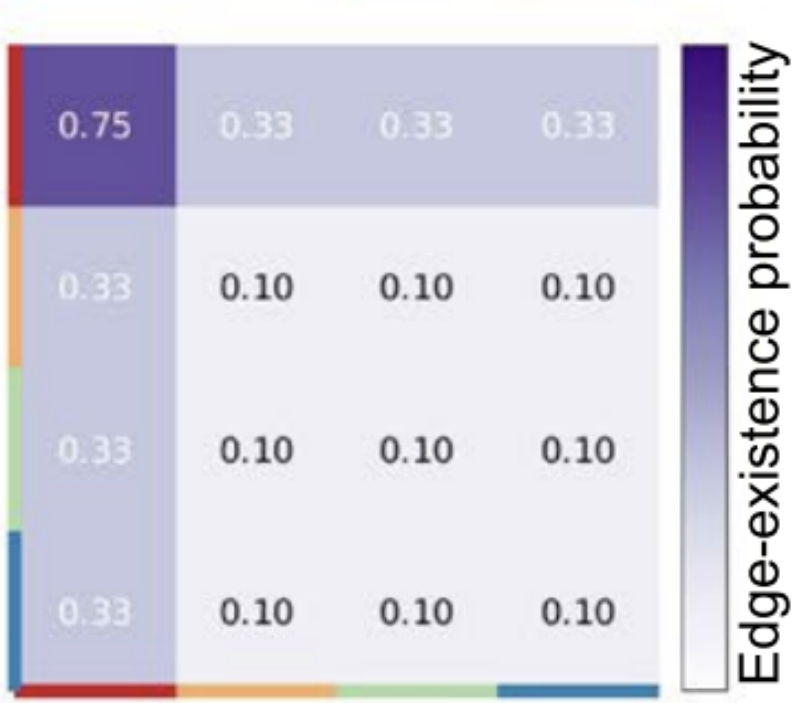


Figure: One cluster with lots of connections within and to which many nodes connect

Answer


0.75
0.33
0.33
0.33
0.33
0.10
0.10
0.10
0.33
0.10
0.10
0.10
0.33
0.10
0.10
0.10
Edge-existence probability

Learning Parameters

Let $\mathbf{Z}$ by an $N \times K$ indicator matrix with $Z_{ik} = 1$ if a node is assigned to community k and $Z_{ik} = 0$ otherwise [1]. $\mathbf{A}$ is our binary $N \times N$ adjacency matrix.

$$A_{ij} \sim \text{Bernoulli}(\boldsymbol{\theta}_{z_i, z_j}) \tag{3}$$

The complete data log-likelihood of the observed graph ($\mathbf{A}$) and the node-to-community assignments ($\mathbf{Z}$) can be written as,

$$\log P(\mathbf{A}, \mathbf{Z}) = \log P(\mathbf{Z}) + \log P(\mathbf{A} \mid \mathbf{Z}) \tag{4}$$

[1] $\mathbf{z}_i$ gives the cluster assignment of node i.

SBM Complete Data Log Likelihood

$$\log P(\mathbf{A}, \mathbf{Z}) = \sum_i \sum_q Z_{iq} \log \alpha_q + \frac{1}{2} \sum_{i \neq j} \sum_{q,l} Z_{iq} Z_{jl} \log b(A_{ij}, \theta_{ql}) \tag{5}$$

- α_q is the probability (in general) of being in community q.
- $b(a, \pi) = \pi^a (1 - \pi)^{1-a}$

SBM Complete Data Log Likelihood, Continued

$\sum_i \sum_q Z_{iq} \log \alpha_q + \frac{1}{2} \sum_{i \neq j} \sum_{q,l} Z_{iq} Z_{jl} \log b(A_{ij}, \theta_{ql})$ can be written completely as,

$$\sum_i \sum_q Z_{iq} \log \alpha_q + \frac{1}{2} \sum_{i \neq j} \sum_{q,l} Z_{iq} Z_{jl} \underbrace{[A_{ij} \log(\theta_{ql})}_{\text{edges}} + \underbrace{(1 - A_{ij}) \log(1 - \theta_{ql}))]}_{\text{no edges}} \tag{6}$$

Fitting Parameters

- I will not go through it here, but you can either use expectation maximization (EM), belief propagation, or MCMC methods.
  - See → `https://arxiv.org/abs/1207.2328` for EM and BP
  - See → `https://journals.aps.org/pre/abstract/10.1103/PhysRevE.89.012804` for MCMC
- The fastest implementation of SBM model parameters that is most readily scalable to large graphs can be found in GraphTool → `https://graph-tool.skewed.de`

Issue

The presented approach is most appropriate for unweighted graphs. Currently, it is more challenging to accommodate edge-weights well without assuming some kind of a distribution on edge weights.

- Any ideas about this?

SBM Applied To Spatial Transcriptomics Dataset

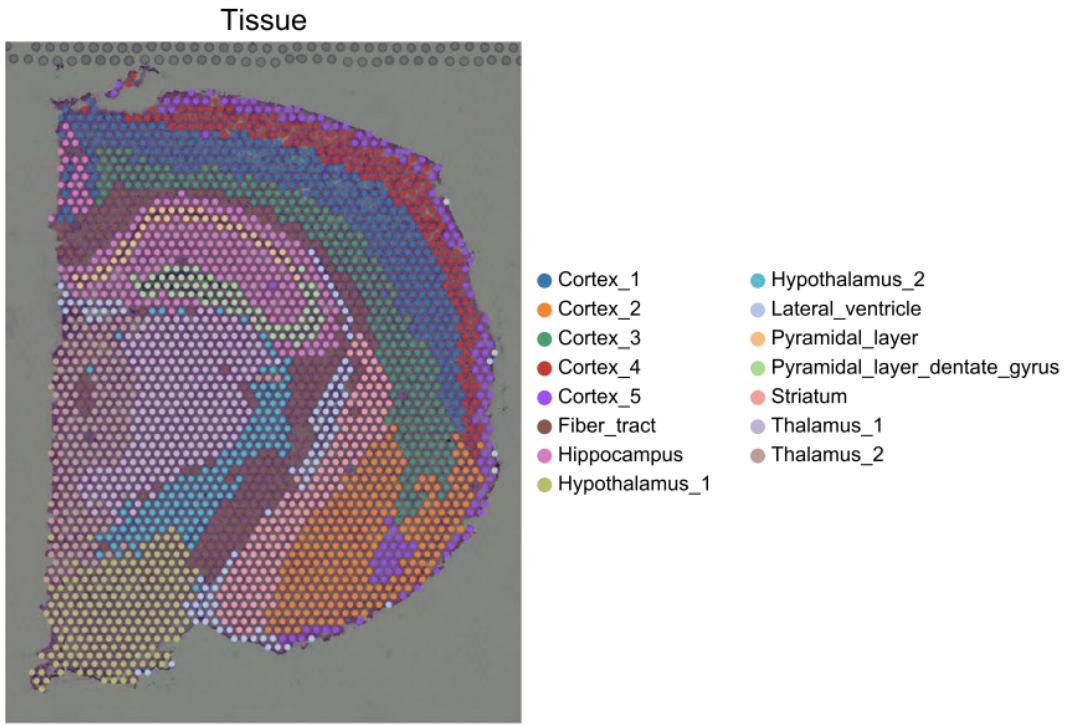
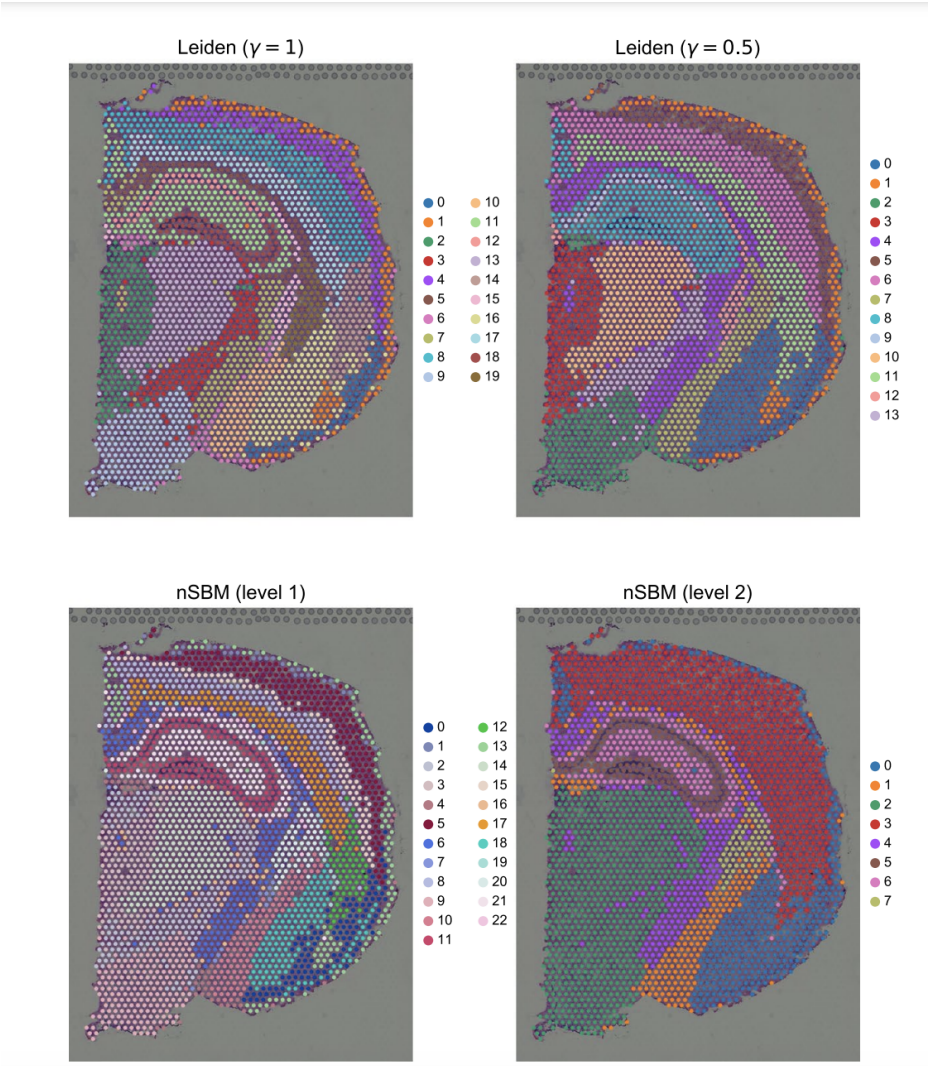


Figure: ground-truth annotation of cells within a coronal section of a mouse-brain from `https://link.springer.com/article/10.1186/s12859-021-04489-7`

Leiden vs SBM



SBM for the Human Microbiome

Nodes $\rightarrow$ microbial species, edges $\rightarrow$ interactions at bodysites. Infer multiple SBMs, where each SBM characterized several body sites according to similar microbial species interactions.

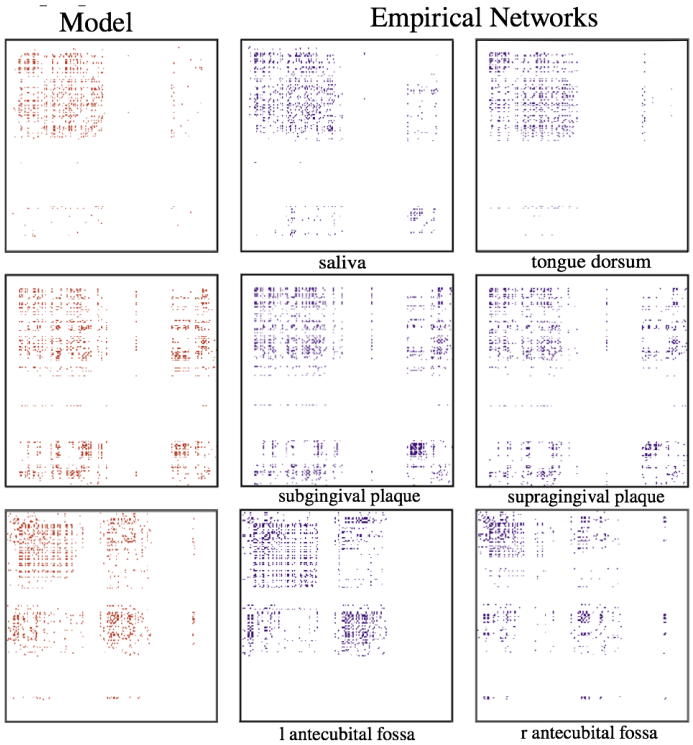


Figure: from Stanley *et al.* IEEE TNSE. 2016.

Affiliation Model for Community Structure

- This is a *soft* clustering approach where overlapping communities are allowed
- Instead of learning a hard node-to-community partition, we learn a node-to-community *propensity*

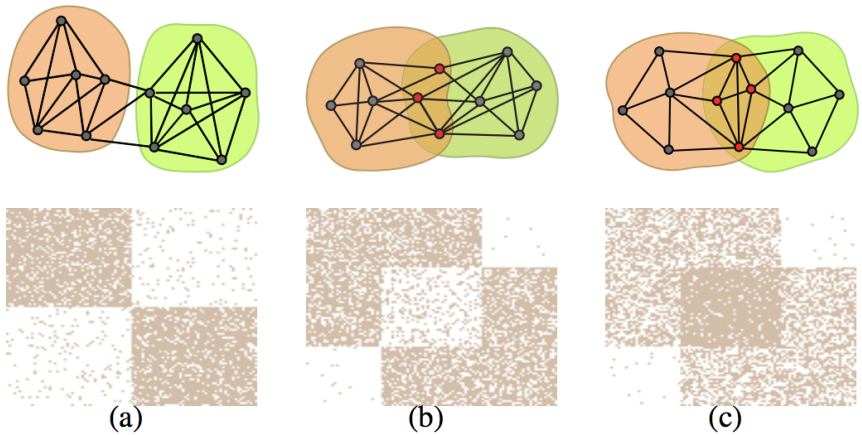


Figure: from Yang *et al.* ICDM. 2012

'BigClam' Approach to Overlapping Communities

Model the existence of an edge between nodes u and v based on the inner product of propensities, F_u and F_v.

DEFINITION 1. *Let F be a nonnegative matrix where F_{uc} is a weight between node $u \in V$ and community $c \in C$. Given F, the* BIGCLAM *generates a graph $G(V, E)$ by creating edge (u, v) between a pair of nodes $u, v \in V$ with probability $p(u, v)$:*

$$p(u, v) = 1 - \exp(-F_u \cdot F_v^T), \tag{1}$$

where F_u is a weight vector for node u ($F_u = F_{u\cdot}$).

Figure: from Yang and Leskovec. WSDM. 2013.

Finding the Optimal F_u

The log-likelihood of the graphs given the learned propensities, $\mathbf{F}_u$ can be expressed as follows. The authors optimize each $\mathbf{f}_u$ by fixing $\mathbf{f}_v$.

$$l(F_u) = \sum_{u \in \mathcal{N}_u} \log(1 - \exp(-F_u F_v^T)) - \sum_{v \notin \mathcal{N}(u)} F_u F_v^T \tag{7}$$

See their paper for more details. `https://cs.stanford.edu/people/jure/pubs/bigclam-wsdm13.pdf`.

Edge Probability vs Number of Shared Memberships

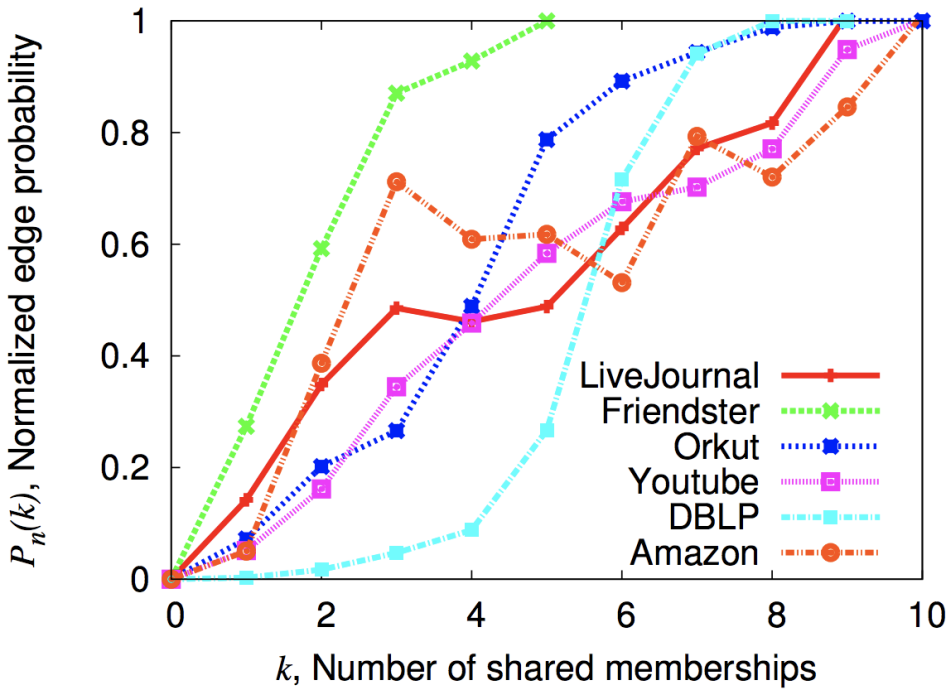


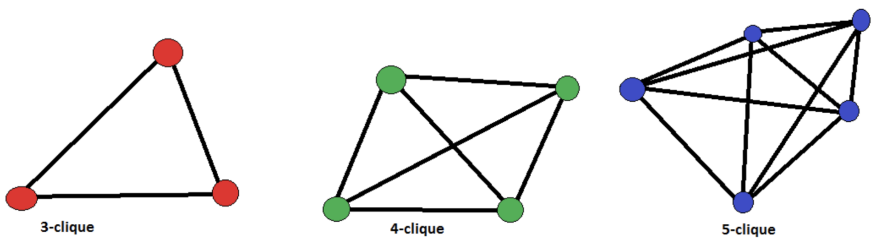
Figure: from Yang and Leskovec. WSDM. 2013.

The Future of Graph Partitioning

- I think overall, this is a pretty solved problem.
- Graph embedding people are still playing, but the partitions of nodes don't look super different than Louvain....
- In one new direction, we want to prune away some of the graph and explore structure without dealing with the whole graph.
- We can also think about defining communities based on motifs, or higher-order edge connections

Higher Order Idea One: Counting Cliques

- **Definition:** A k-clique is set, S, of k nodes, such that all pairs in S are connected by an edge.
- In some applications, you might want to study cliques. Or maybe you want to study the behavior of cliques as you make decisions in constructing your graph.
- Maybe it is easy to count the number of triangles in the graph but as k gets larger, this can become quite difficult
- If this is interesting to you, there is some very nice theoretical work for approximating the number of cliques in a graph `https://arxiv.org/pdf/1611.05561.pdf`



For those who count cliques

The Turan Shadow algorithm is implemented in Julia
`https://github.com/nassarhuda/TuranShadow.jl`