Comp790-166: Computational Biology

Lecture 3

January 26, 2021

Today

- Graph Partitioning (Community Detection)
  - Modularity Based Methods (Louvain and Leiden)
  - Benchmarking module detection methods in biological networks
  - Stochastic Block Model
  - Affiliation Model
  - Application of SBM in biology.

Community Detection Illustrated

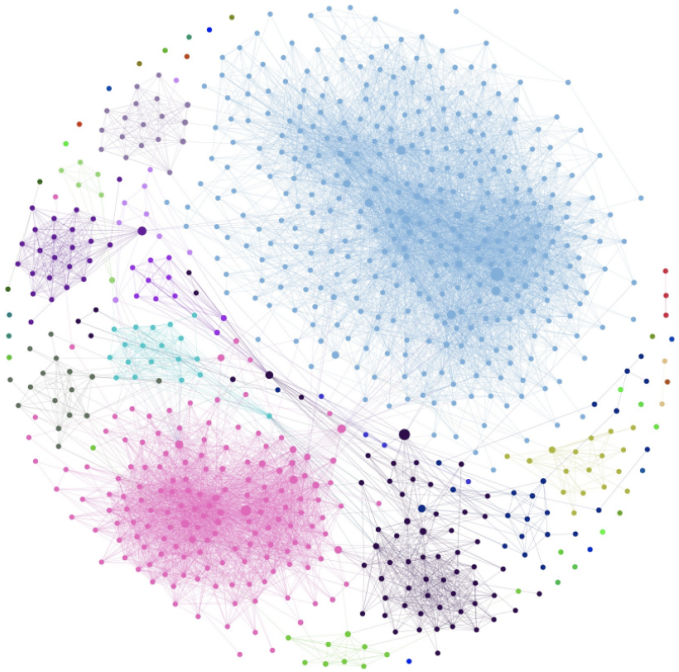

Figure: The high-level goal is to partition nodes into coherent node-subsets (communities), such that nodes within a community have on average more within group edges than between-group edges.

Community Detection (Graph Partitioning) is just Clustering

- Instead of your standard clustering problem in a matrix of N objects with P features, our input here is an adjacency matrix, $\mathbf{A}$, where we want to cluster nodes based on similarities in their connectivity patterns.
- Most community detection optimization problems seek a hard partition (each node assigned to a single community)
- There are some variants that learn a soft partition, or a propensity or probability that each node is assigned to each community.
- Optimization for this problem can come in many flavors

Optimization Approaches

- Quality Function with a Null Model + Heuristic for Optimization ←
- Probabilistic and Likelihood Optimization ←
- Spectral Clustering Methods
- Most recently: graph embedding + clustering on embedding

Why Might a Person Waste Time Partitioning a Graph?

Build a graph between cells based on marker expression and partition into cell-populations. Members of a cell-population should be phenotypically similar and therefore express the same markers.

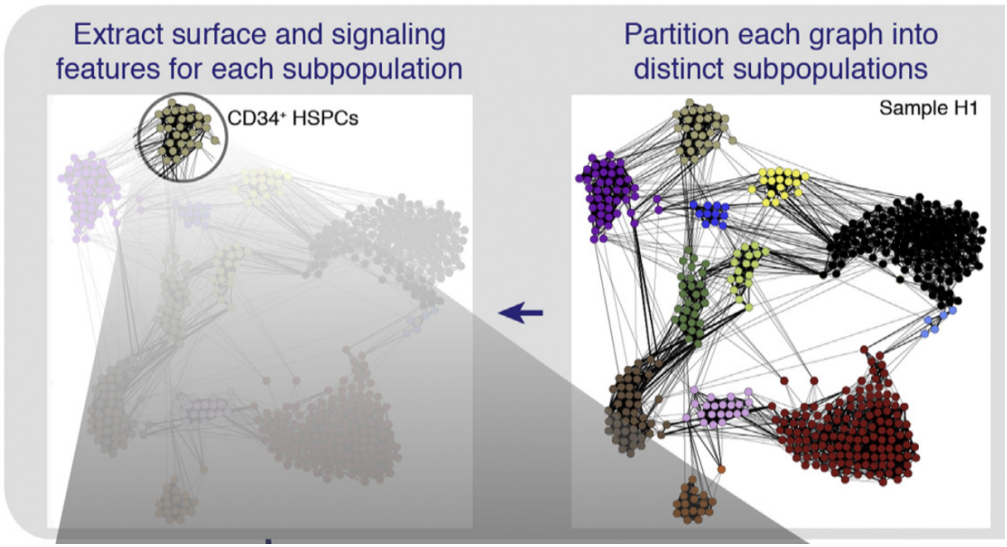

Figure: from Levine *et al.* Cell. 2015. This is the PhenoGraph algorithm.

PhenoGraph Uses Modularity Based Maximization

- **Intuition:** We first specify a null model of a network with no structure about the probability of two nodes being connected.Then we find the partition that is maximally different from this null model.
- **A Simple Null Model:** An easy way to think about the probability of two edges being connected is based on some function of their degree.
  - Consider the null model, $\frac{k_i k_j}{2M}$
  - k_i and k_j give the degrees of nodes i and j
  - M (as always in our notation) is the total number of edges, or the sum of all edge weights.
- **Configuration Model:** This is known as the configuration model, or fixing the degree sequence and connecting nodes at random.

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.

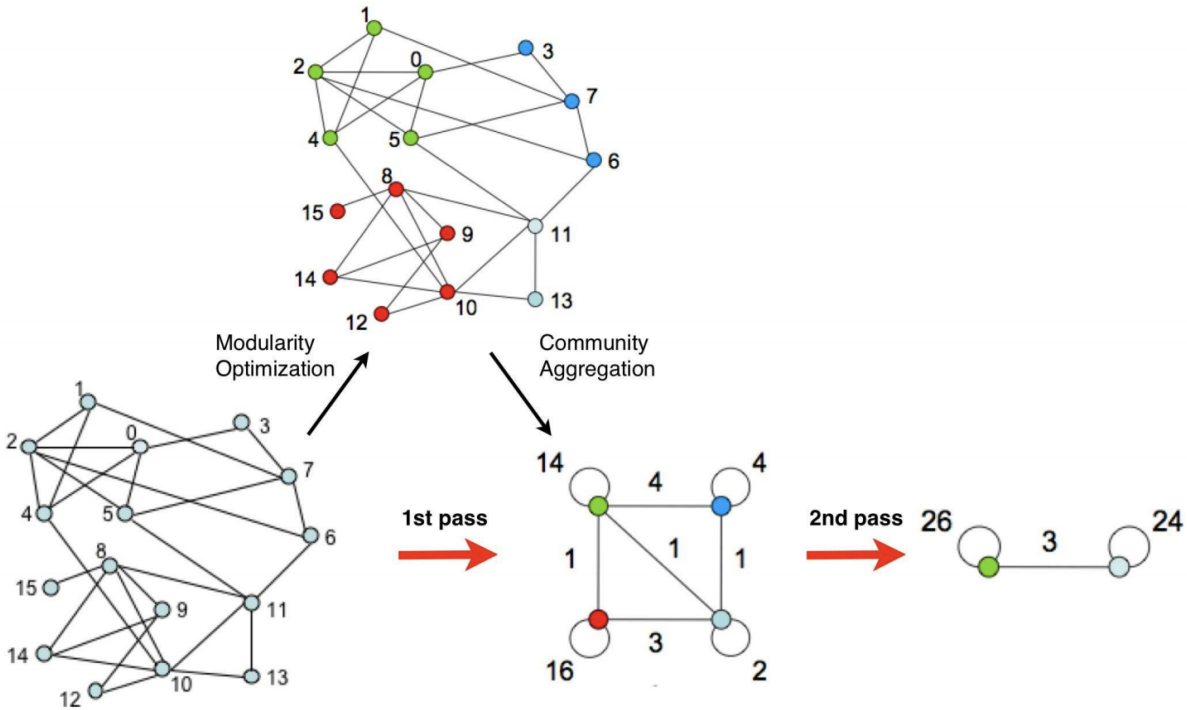

Figure: from Blondel *et al.* Journal of Statistical Mechanics. 2008.

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

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.

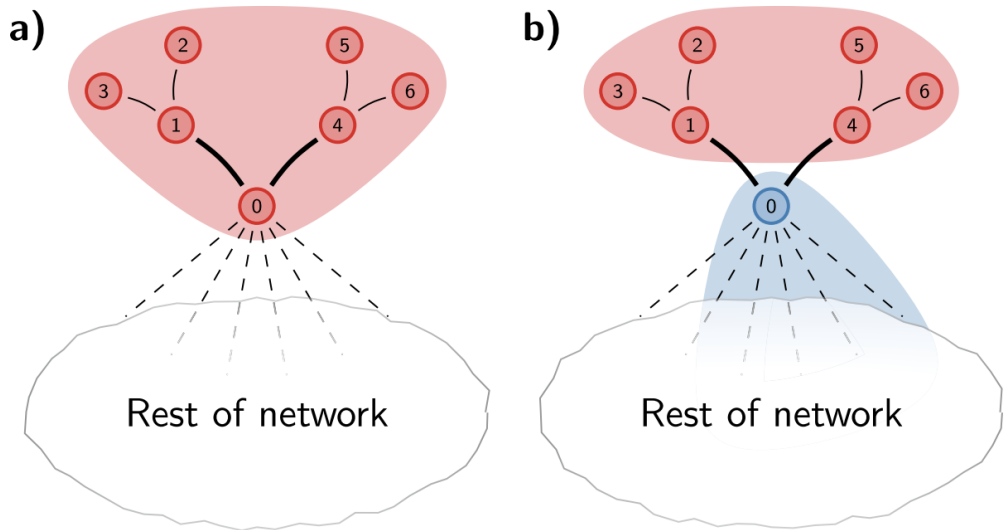

Figure: from Traag *et al.* Scientific Reports. 2018.

Overview of Leiden

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

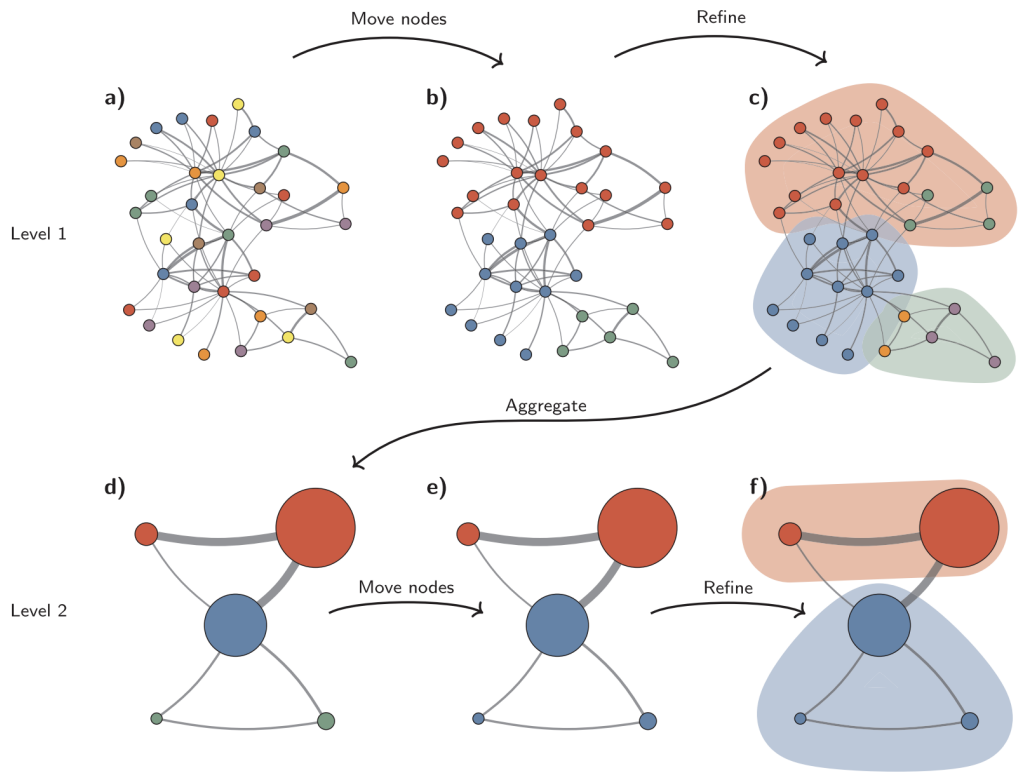

Figure: from Traag *et al.* Scientific Reports. 2018.

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?

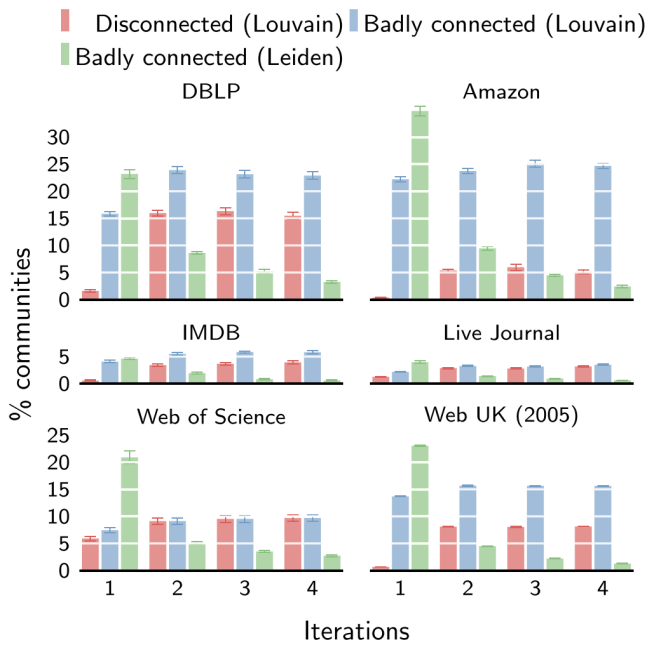

Figure: from Traag *et al.* Scientific Reports. 2018.

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`

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.

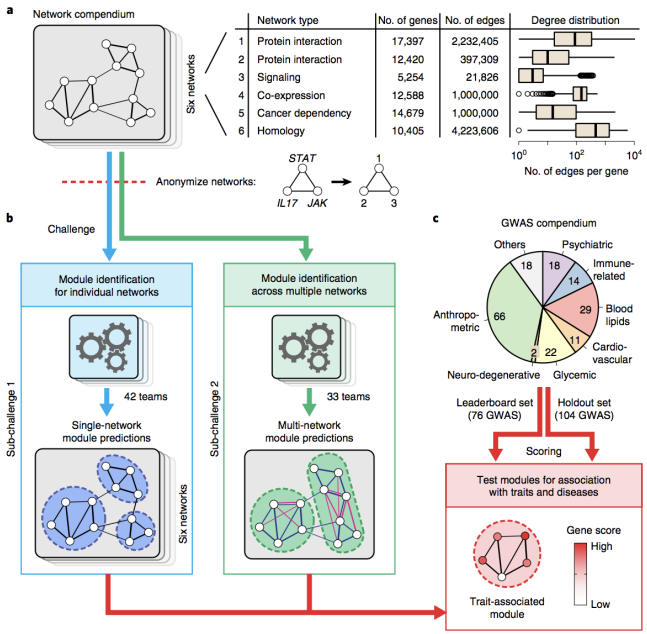

Figure: from Choobar *et al.* Nature Methods 2018.

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.

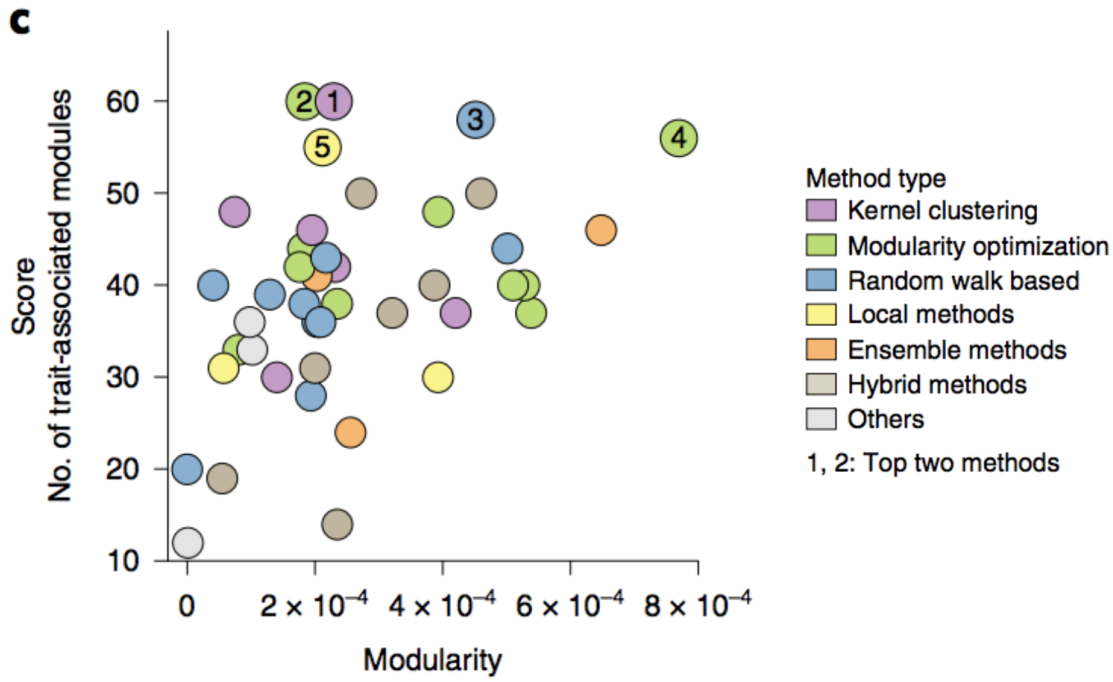

Figure: from Choobar *et al.* Nature Methods 2018.

Ranking Methods Based on Biological Relevance.

'M1' is a variant of what we have seen with Louvain. Interested readers can read more here, `https://arxiv.org/abs/physics/0703218`

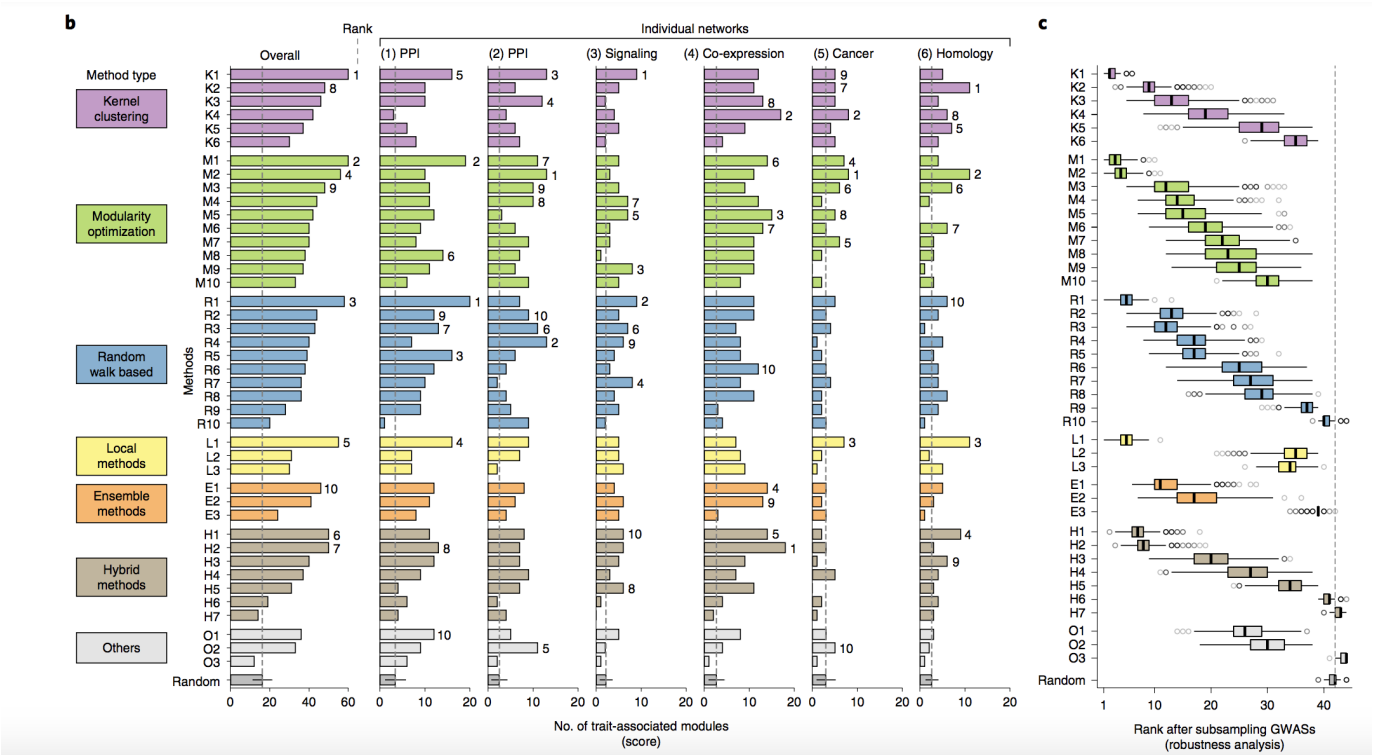

Figure: from Choobar *et al.* Nature Methods 2018.

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
  - θ, a matrix of between-community edge probabilities
  - $\mathbf{z}$, a vector of node-to-community assignments

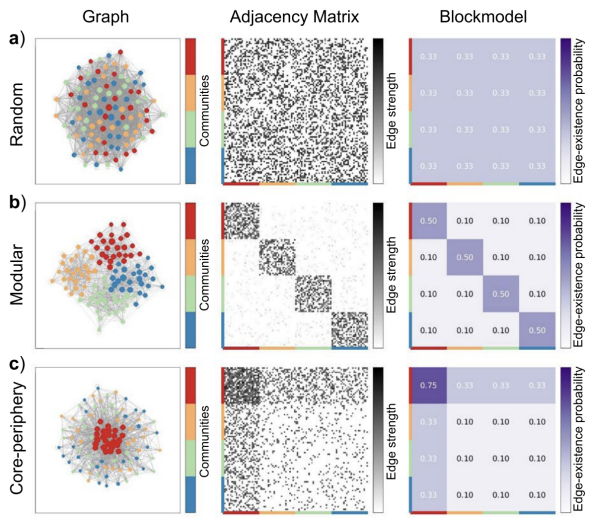

Figure: from Fastkowitz *et al.* Scientific Reports. 2018.

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. $\mathbf{A}$ is our binary $N \times N$ adjacency matrix.

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

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

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

SBM Complete Data Log Likelihood

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

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

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`

An Application of the SBM in Single-Cell Data

Using single-cell RNA-seq data, the authors build a graph between cells based on gene expression and inferred phenotypically homogeneous groups of cells with an SBM. The authors claim that the modularity-based optimization approach can prevent the identification of small communities.

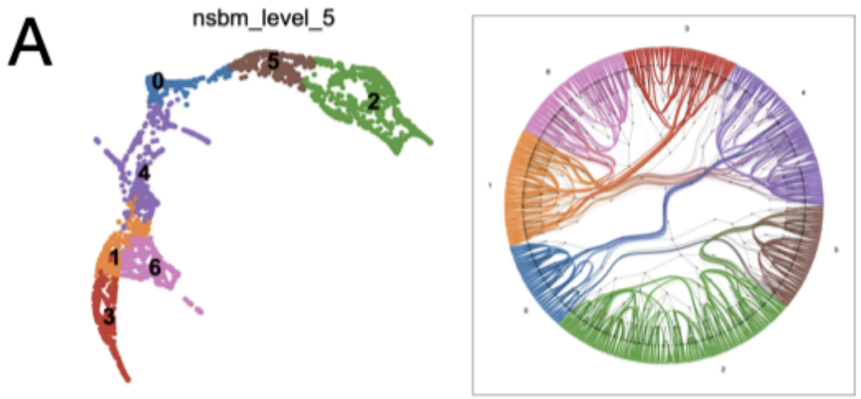

Figure: from
`https://www.biorxiv.org/content/10.1101/2020.06.28.176180v2.full`

SBM in the Analysis of the Human Microbiome

Here nodes are different microbial species and their interactions collected from different bodysites. Learning an ensemble of stochastic block models, where each member of the ensemble characterized several body sites according to similar microbial interactions.

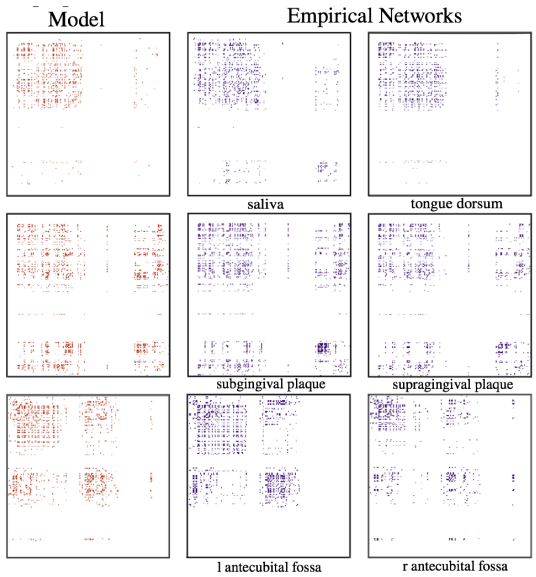

Figure: from Stanley *et al.* IEEE TNSE. 2016.

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*

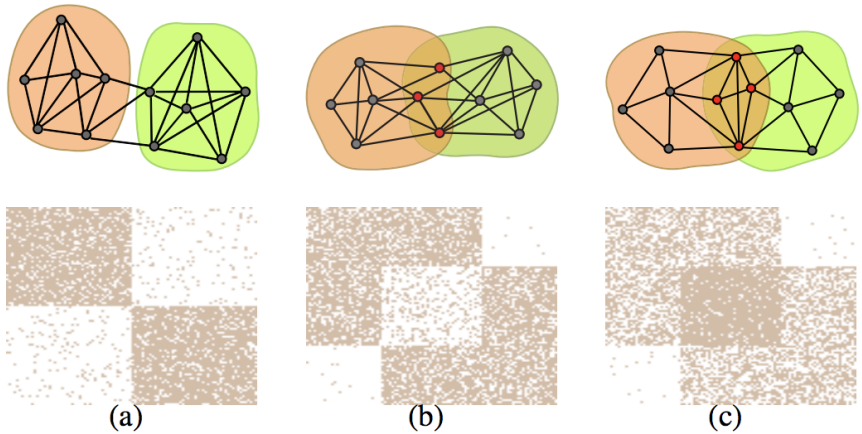

Figure: from Yang *et al.* ICDM. 2012

'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. WSBM. 2013.

Finding the Optimal F_u

The authors use a block coordinate ascent approach where they fix all F_v to find the best F_u.

$$l(F_u) = \sum_{u \in \mathcal{N}_u} \log(1 - \exp(-F_u F_{v^T})) - \sum_{v \notin \mathcal{N}(u)} F_u F_v^T \tag{6}$$

They can use gradient ascent to find each F_u. See their paper for more details. `https://cs.stanford.edu/people/jure/pubs/bigclam-wsdm13.pdf`.

Next time

- Network embeddings, according to either nodes or edges (you can also find ‘communities‘ in embedding space.
- Moving towards graph signal processing
- We are almost ready to start bioinformatics.....