Community detection and k-cores decomposition

You are given three small **social networks with known partitioning** of nodes.

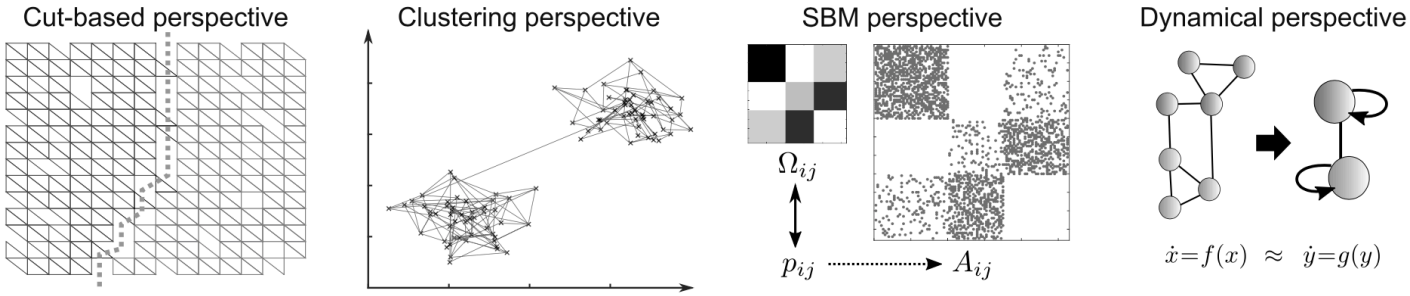
- Zachary’s karate club network (karate.net, 2 clusters)
- Lusseau’s bottlenose dolphins network (dolphins.net, 2 clusters)
- Davis’s southern women network (women.net, 2 to 4 clusters)

Later you will be studying also selected larger **networks with node labels**.

- Game of Thrones character coappearance network (got-appearance.net, characters)
- Human diseasome network by common symptoms (diseasome.net, diseases)
- Conflicts and alliances between world nations (wars.net, countries)
- Ingredients network by shared compounds (ingredients.net, ingredients)

All networks are available in Pajek format.

Browse CDlib library for implementations of **community detection or graph partitioning** algorithms. Select an algorithm that you will be using for the exercises below. Some of the most popular algorithms are optimization of modularity such as the Louvain and Leiden algorithms, map equation algorithm called Infomap, fast label propagation algorithm, hierarchical clustering based on edge betweenness, stochastic block models etc.



I. Small networks with known sociological partitioning

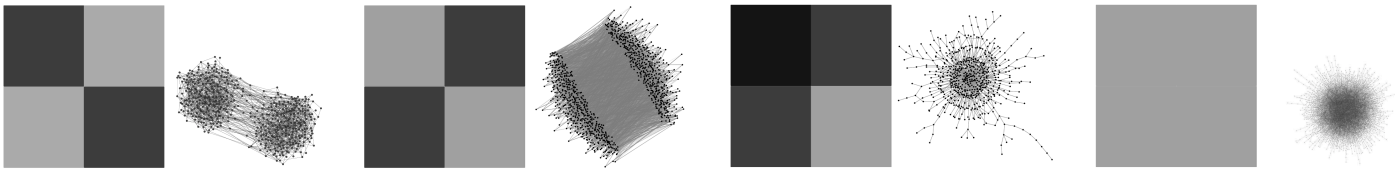
(starter) Apply the selected algorithm to **small social networks** and test whether **communities coincide with sociological partitioning** of these networks. Since most algorithms are not deterministic, you should apply the algorithm to each network multiple times and compare the partitions using a standard measure such as the normalized mutual information or the adjusted Rand index. Since these networks are very small, you can also visualize communities.

II. Larger networks with labels associated with nodes

Apply the selected algorithm to **larger real networks** and test whether **communities provide good decomposition or abstraction** of the networks. Since most algorithms are not deterministic, you should apply the algorithm to each network multiple times and examine the labels of the nodes in different communities. For simplicity, you may rather compute some standard quality function such as modularity or examine only the nodes in the largest community.

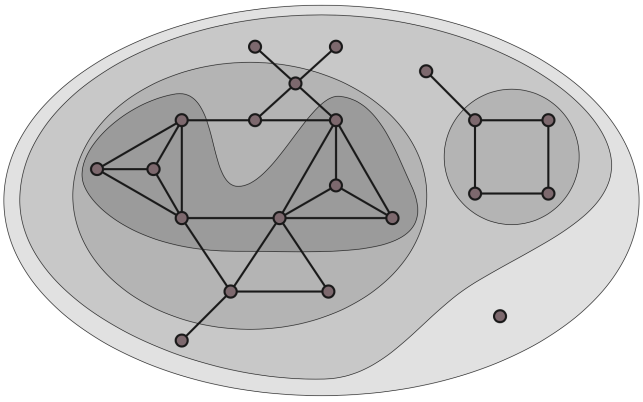
III. Random graphs with no mesoscopic structure

(tentative) Apply the selected algorithm also to **Erdös-Rényi random graphs** with **no community or other mesoscopic structure** and test whether the algorithm can recognize this. Apply the algorithm to random graphs with increasing average degree $\langle k \rangle$ and test for which values of $\langle k \rangle$ the algorithm no longer detects any structure.



IV. k-cores decomposition of real networks

1. Recall the following **algorithm for computing network k-cores** for a given k. Starting with the original network, iteratively remove nodes with degree less than k. When no such node remains, connected components of the resulting network are the k-cores.



2. *(tentative)* Implement the algorithm and compute all **k-cores of larger networks** above. Remember that a k-core is always a subset of a $k-1$-core. What is the maximum value of k denoted k_{main} for which there exists at least one k-core?

3. For each network, print out the **labels of nodes in k_{main}-cores** and try to interpret the results.