Kernighan-Lin, Graph Distance Metrics

858L

Kernighan-Lin Graph Partitioning

Kernighan-Lin Graph Partitioning

Problem. Divide a weighted graph with 2n nodes into two parts, each of size n, to minimize the sum of the weights crossing the two parts.

> You could just use minimum cut if you didn't have the restriction to each part being of size n.

Idea (Kernighan-Lin, 1970): start with some partition that satisfies the size requirement and repeatedly swap nodes between the partitions.

Kernighan-Lin More Detail

Divide the network into 2 parts A, B of equal size arbitrarily.

Repeat until no more vertices are left:

Select $a_i \in$ A, $b_i \in$ B, such that the reduction in cost is as large as possible and neither a_i, b_i has been chosen before

Swap a_i and b_i

Let C_i be the cost of the partition after swapping a_i,b_i

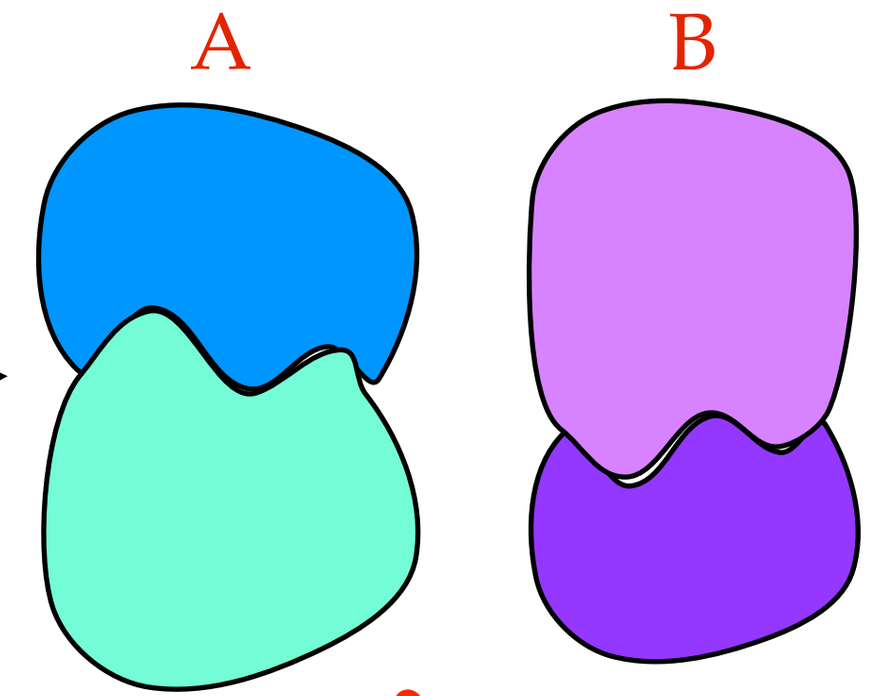
Return (A',B') corresponding to the smallest C_i observed.

While cost continues to be reduced

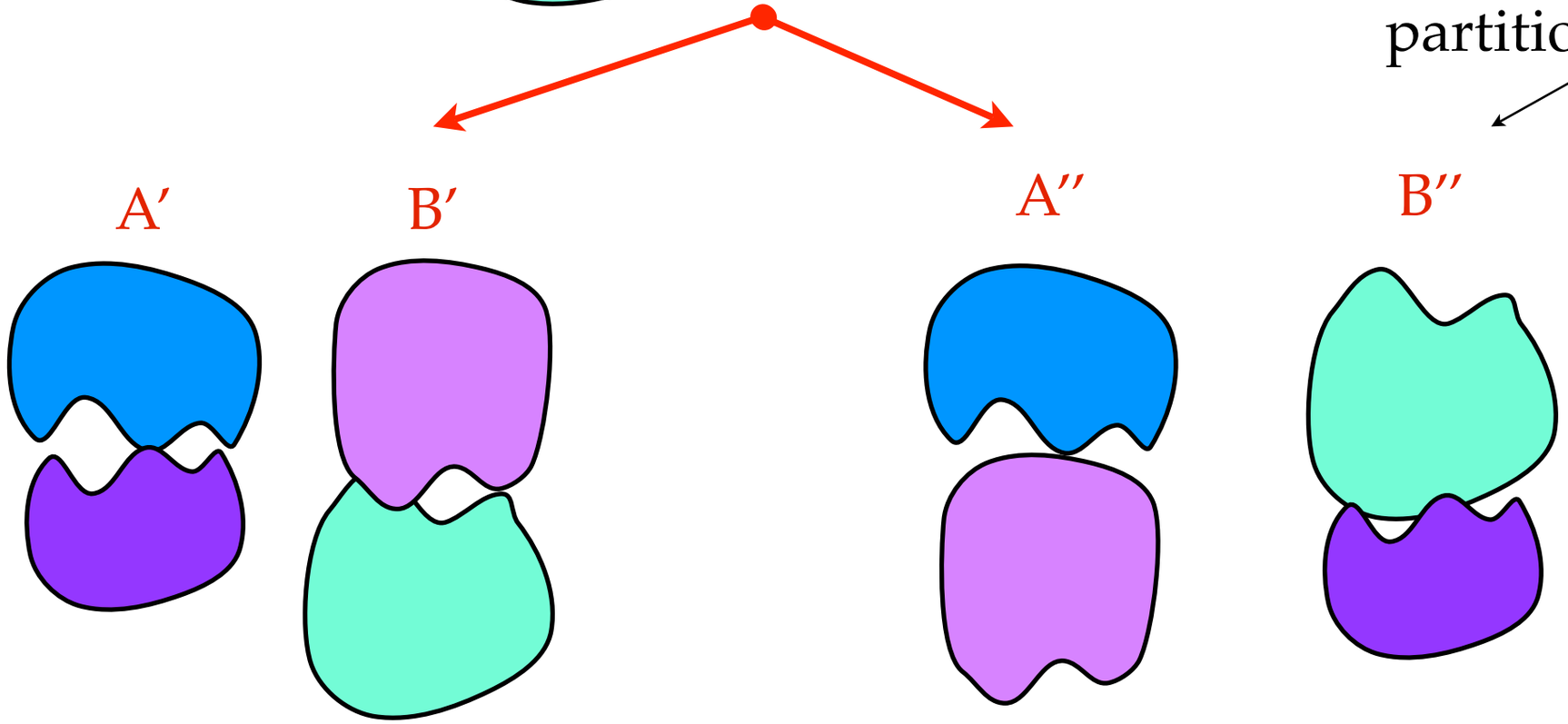
Call using the returned partition as the new starting point

Improving this Solution

Find the optimal partitions in A and B **separately**



Run KL procedure on all 3 of these starting partitions



Three KL Extensions

(1) Divide into partitions of unequal (but known) sizes:
Start with a partition that satisfies the sizes you want.
Stop when all the nodes on the smaller side have been swapped.

(2) Divide into 2 partitions such that one has $\geq n_1$ nodes and the other has $\leq n_2$ (where $n_1 + n_2 = n$).
Left as an exercise or exam problem, or whatnot.

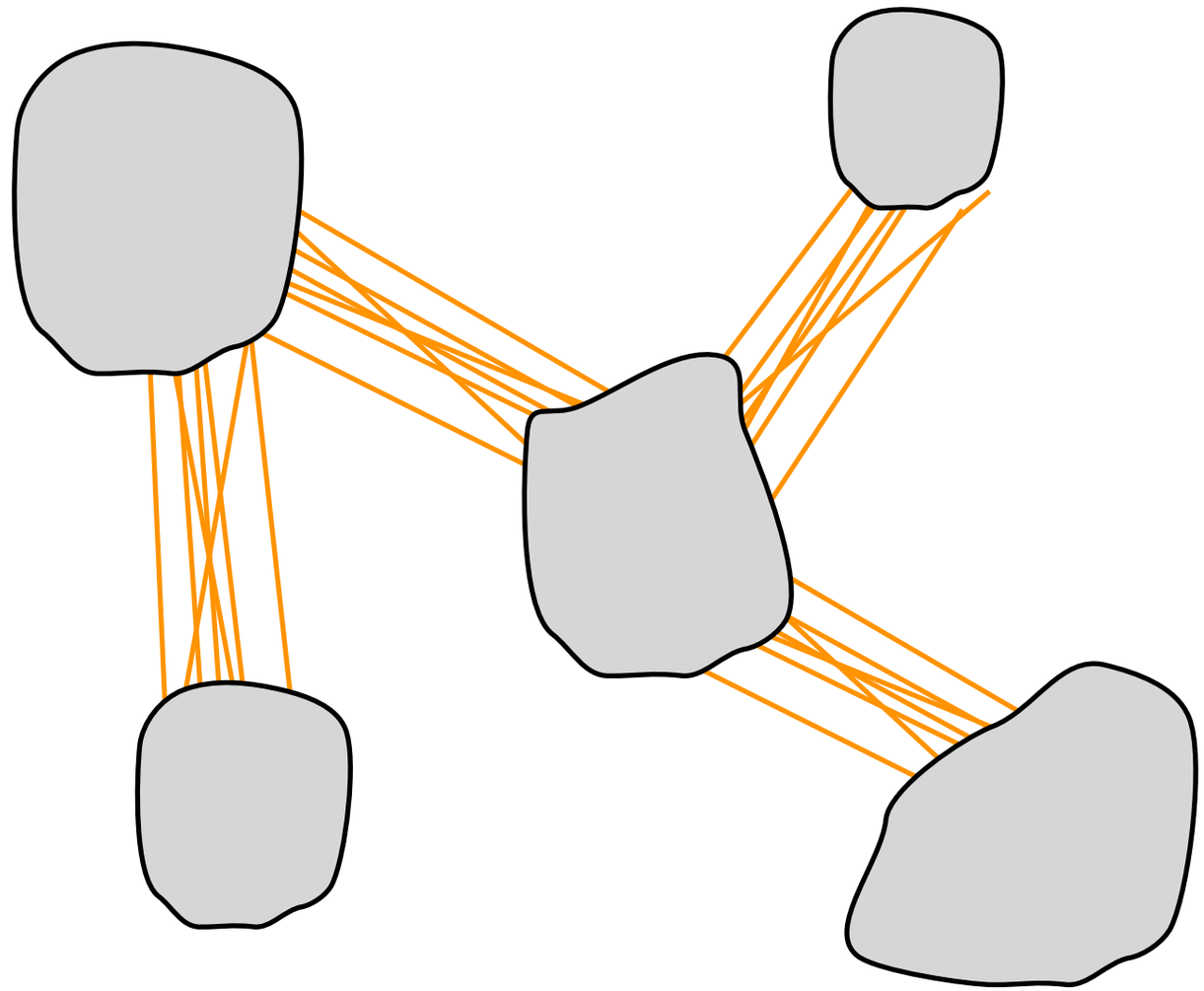
(3) What if you have node weights w_u and want $\sum w_u$ for $u \in A$ to equal some n_1?
If w_u are integers, replace u with a clique of w_u nodes, connected by very high weight edges.

KL Into ≥ 3 partitions

Start with a partition that satisfies your k size requirements $n_1, n_2, ..., n_k$

Apply the 2-part procedure between every pair of parts (n choose 2 times)

Repeat the above step until no improvement is obtained.



Getting a starting partition

Method 1. Suppose you want k partitions.

Let $k = k_1k_2k_3 \cdot \ldots \cdot k_m$

Divide the graph into k_1 parts (starting, say from an arbitrary split)

Divide each of those k_1 parts into k_2 parts

⋮

and so on.

If the k_i are small (say k is a power of 2) then as long as we're OK at getting a 2-split, we get a good k-split.

Method 2. Suppose you want k partitions in a graph of kn nodes. Use the 2-part algorithm to find a $(n, k(n-1))$ split. Let the n-sized set be one part, repeat.

Clustering Using Graph Distances

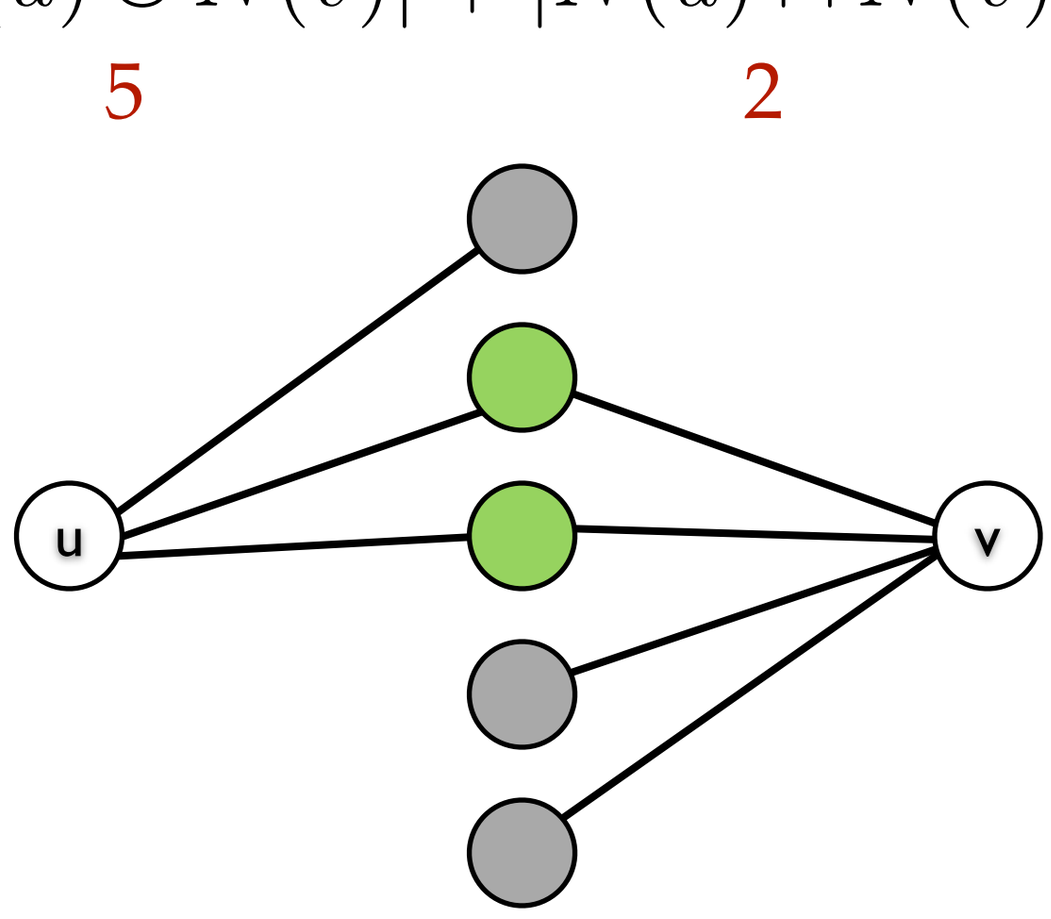
Distance Notions on Graphs

- Apply standard clustering algorithms, need to define d(u,v) as ``distance'' between nodes u and v.

$$\text{Czekanowski-Dice}(u, v) := \frac{|N(u) \,\Delta\, N(v)|}{|N(u) \cup N(v)| + |N(u) \cap N(v)|}$$

when u and v share no neighbors, they get distance 1.0

when they share all their neighbors, they get distance 0



Shortest Path Metric

- Let G be any undirected, unweighted graph
- Define $d_G(u,v)$ be the length of the shortest path between nodes u and v.

- $d_G(u,v)$ is a metric:
  - $d_G(u, v) \geq 0$
  - $d_G(u, v) = 0$ iff $u = v$
  - symmetric: $d_G(u, v) = d_G(v, u)$
  - triangle inequality: $d_G(u, v) + d_G(v, w) \leq d_G(u, w)$

- Using the shortest path as a distance makes sense.

Shortest path metric problems

- Define d(u,v) as the shortest path distance between u and v
  - Use standard clustering algorithms
- **Problem**: there are many distance ties.

- **Solution**: Arnau et al, 2005:

```
1. Compute distance matrix D
2. Repeat for N trials:
   2.1. Randomly sample D to get a subset S of proteins
   2.2. Agglomeratively cluster S, stopping according to a
        distance threshold
3. real_d(i,j) := fraction of trials for which i and j were
   placed into different clusters.
4. Cluster using real_d
```

Comparing Shortest Path Metrics

- Rives & Galitski, 2003 propose:
  - Similarity between proteins i, j:

$$s_{ij} = 1/\text{shortest\_path\_dist}(i,j)^2$$

  - Represent each protein by the vector $\langle s_i \rangle$: it's "shortest path profile"
  - Use hierarchical agglomerative clustering with the distance between i and j defined as:

$$d(i,j) = \text{correlation}(\langle s_i \rangle, \langle s_j \rangle)$$

- **Idea**: similar proteins will have similar relationships to the rest of the network.

$\mathrm{corr}(s_i, s_j)$

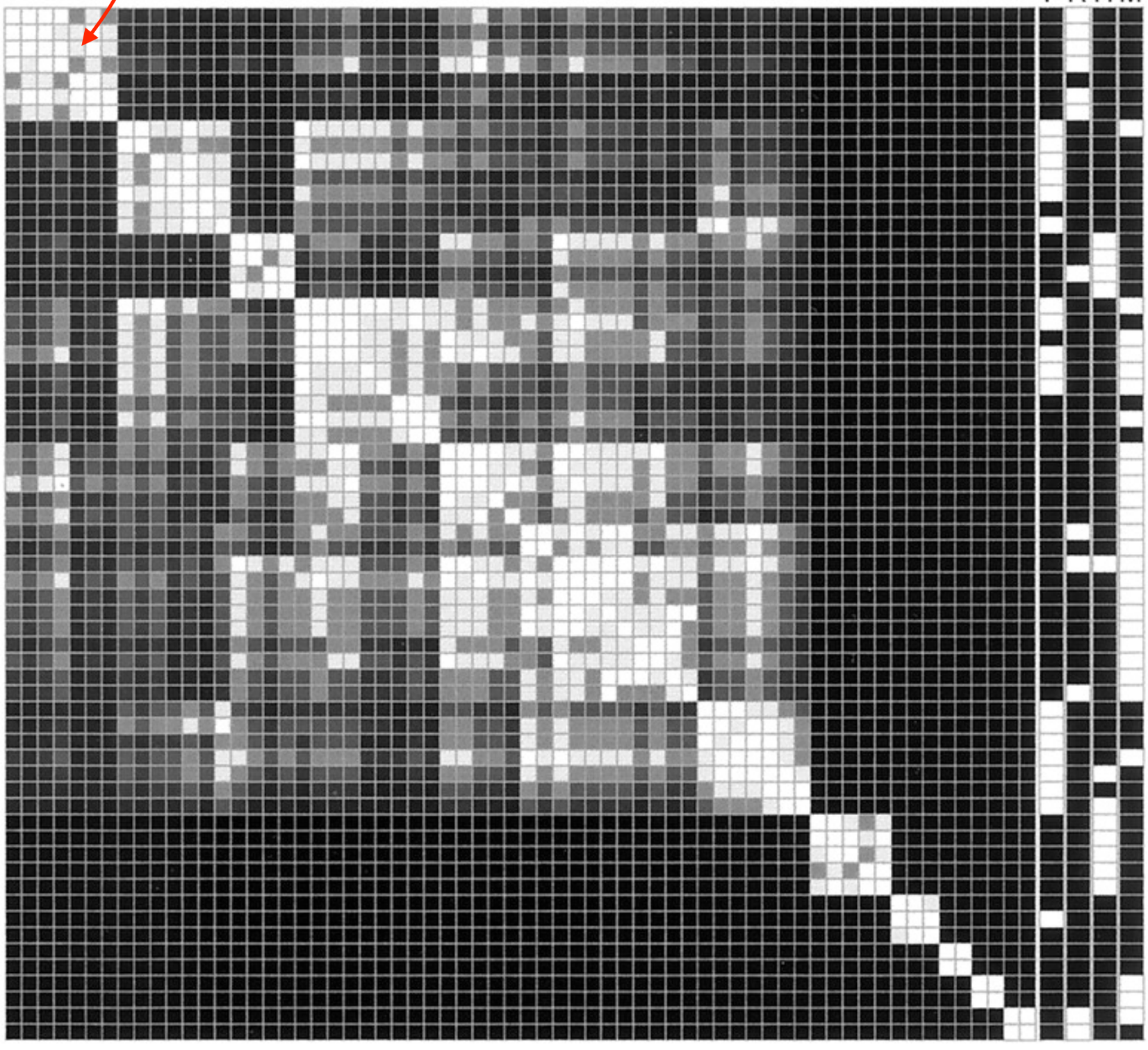
Proteins

P R H M

P,R,H,M represent MIPS-annotated pathways

64 signaling proteins

Proteins



Clustering permutes rows/columns

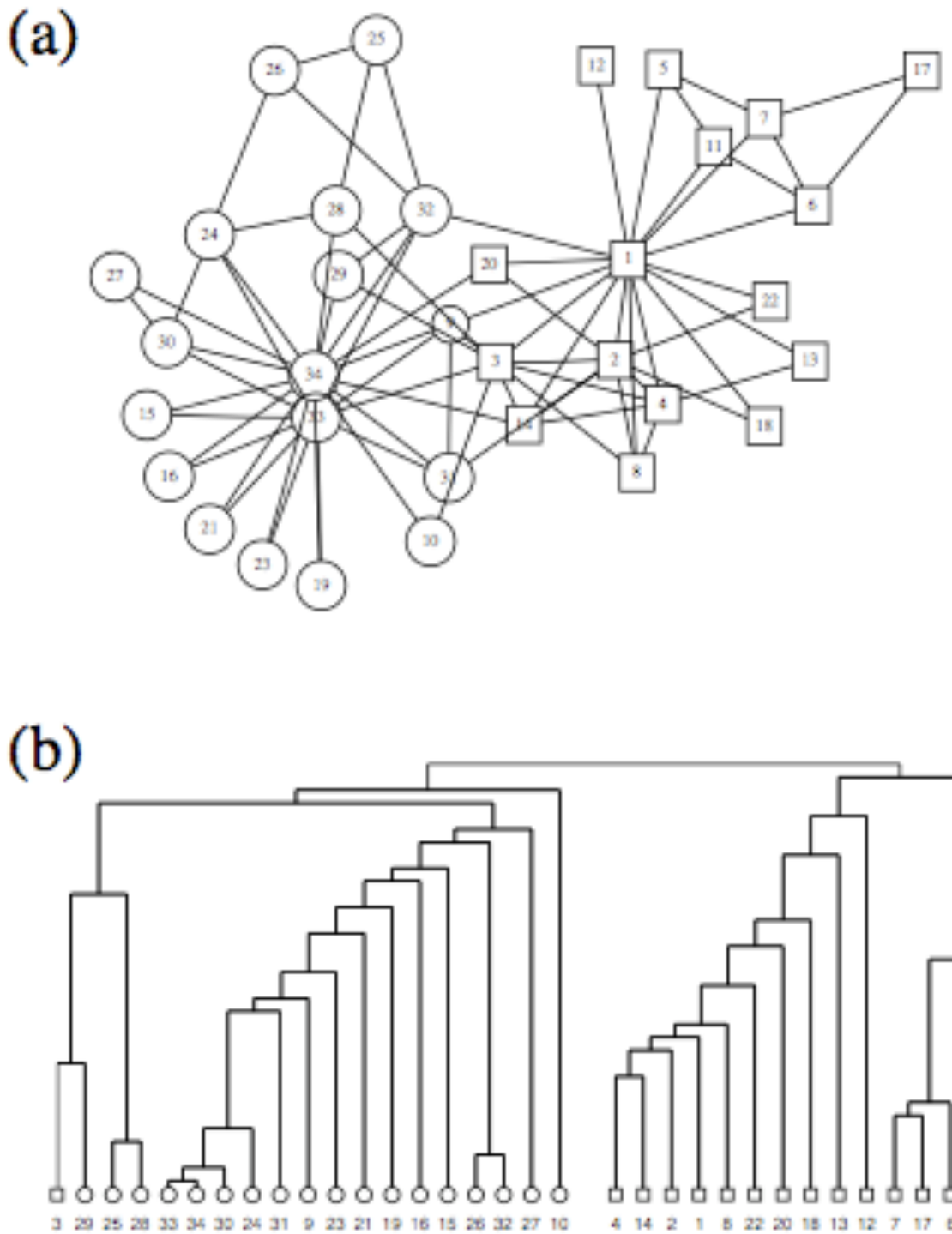
(Rives & Galitsky, 2003)

Girvan-Newman Algorithm

- Edge Betweenness:
  - EB(u,v) := number of shortest paths between two nodes that run through edge {u,v}
  - If there are n shortest paths between a pair of nodes, each is counted with weight 1/n.

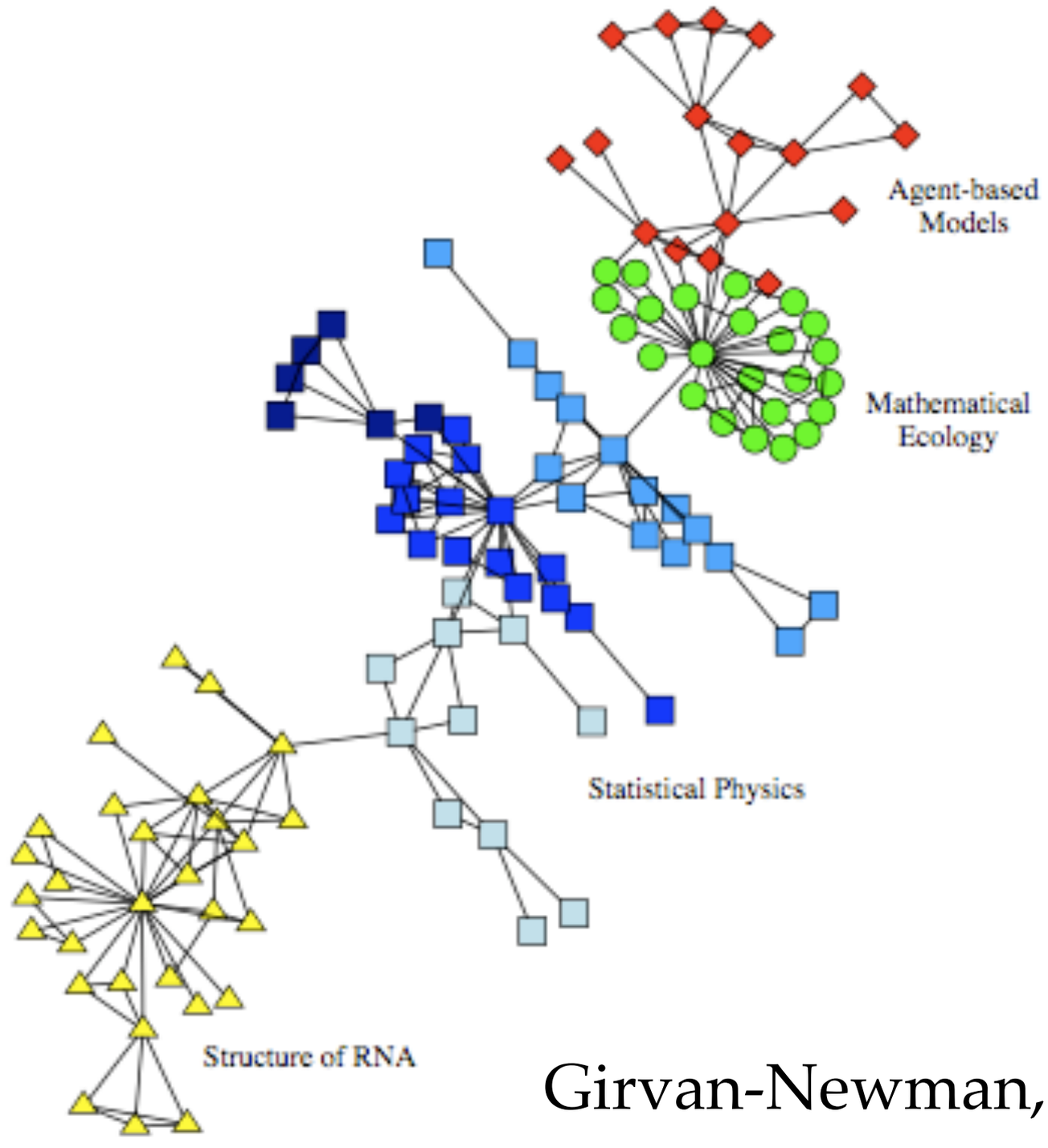
- Girvan-Newman (2002):
  - Repeat until there are no more edges:
    - Remove the edge with the highest betweenness
    - Recalculate the betweenness
  - Clusters are the connected components at some point during the algorithm.

Zachary's karate club



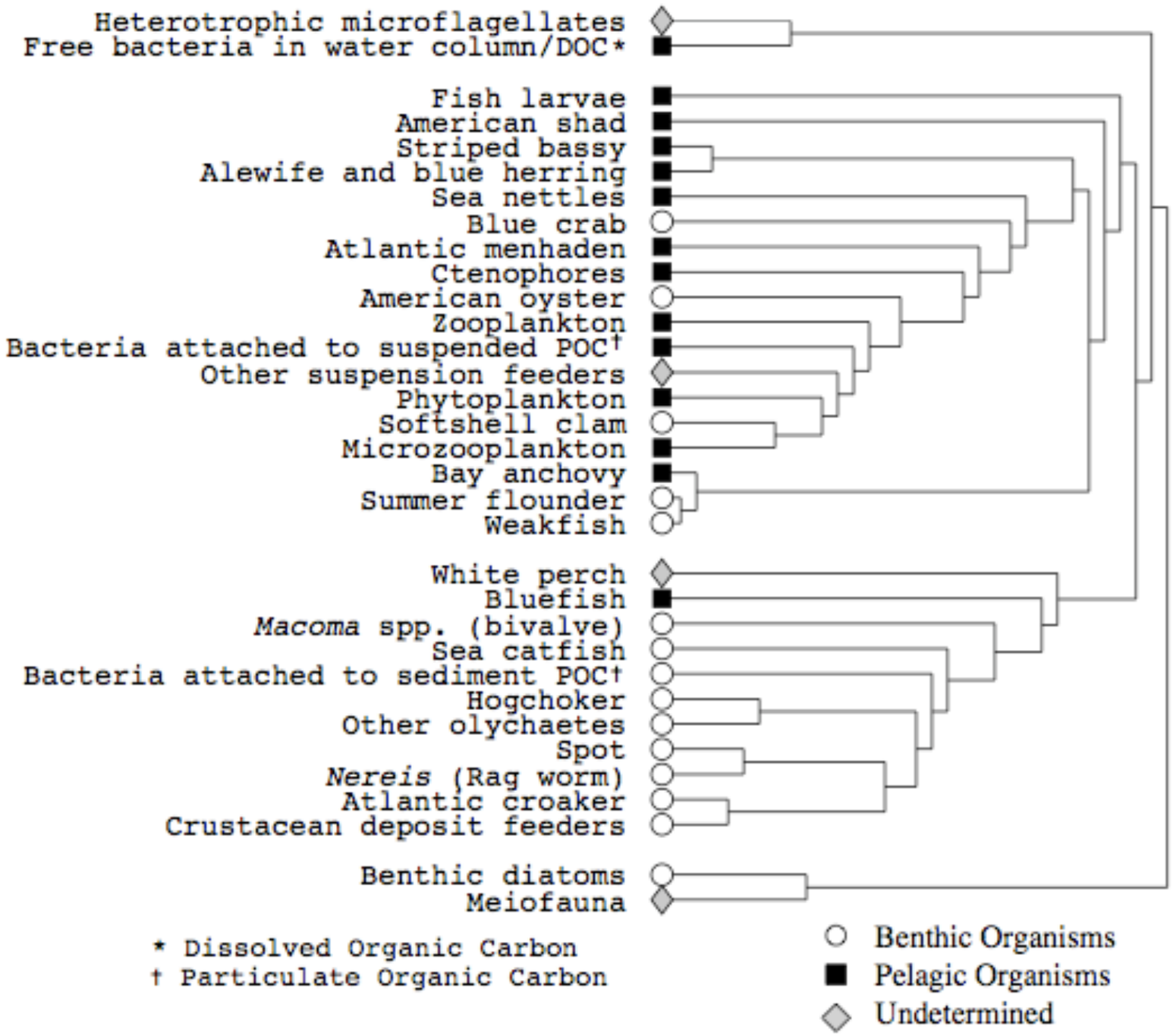
Girvan-Newman, 2002

Santa Fe Collaboration Network



Girvan-Newman, 2002

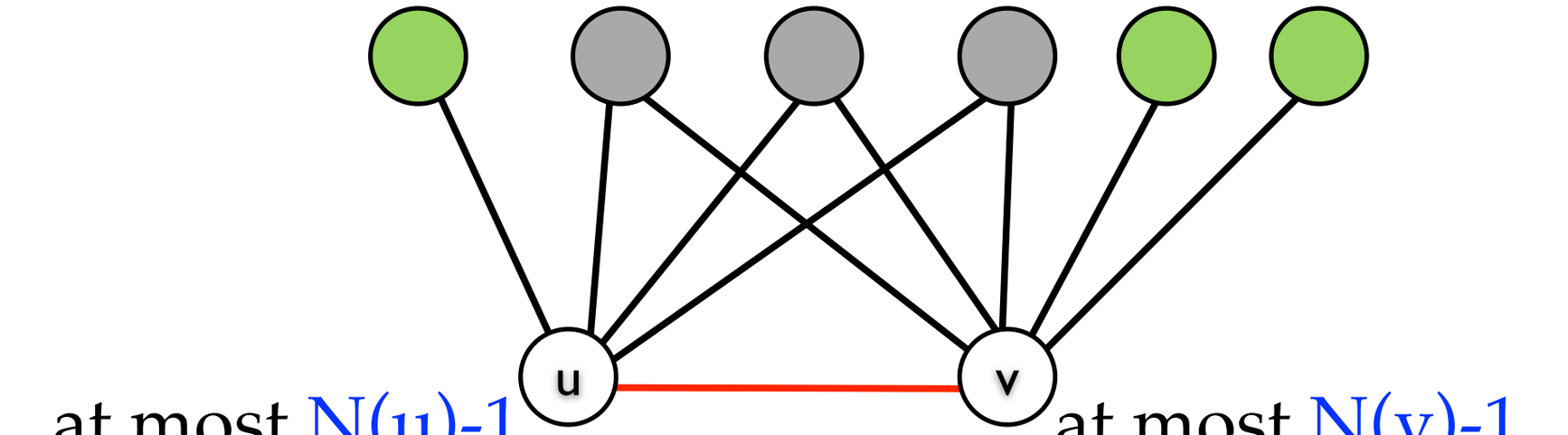
Chesapeake Bay Food Web



Girvan-Newman, 2002

Edge Clustering Coefficient

- Edge Clustering Coefficient = fraction of possible triangles in which an edge is involved:



at most N(u)-1 possible triangles

at most N(v)-1 possible triangles

number of possible shared triangles:
min{N(u)-1, N(v)-1}

$$ECC(u,v) := \frac{\#\text{tri}(u,v)+1}{\min\{N(u)-1, N(v)-1\}}$$

Summary

- Module detection, aka community detection, aka graph clustering, aka graph partitioning is a useful technique for predicting protein function
  - Also useful in other network analysis contexts, such as social networks
- Can define a distance on the network and use a standard clustering technique
  - Shortest path (metric), Shortest path profiles, or % times nodes appear in separate clusters
- Can use edge centrality to define communities
- **Modularity**: a widely used measure of community quality
- Two algorithms for maximizing it: greedy and spectral partitioning-like.
- Kernighan-Lin was a very influential early heuristic, which is still popping up today.