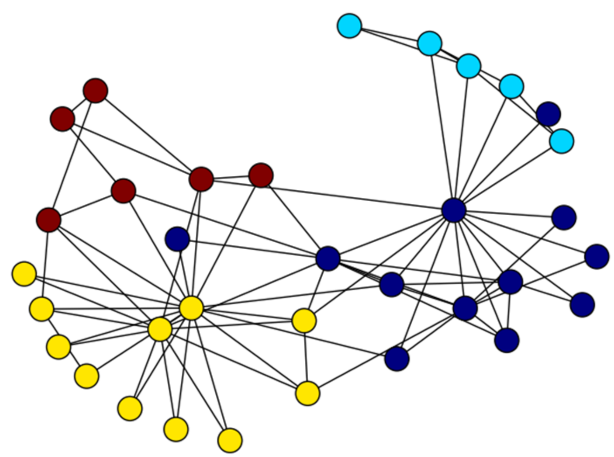
Topological Analysis (2)



Hiroki Sayama
sayama@binghamton.edu

Centralities and Coreness

Centrality measures ("B,C,D,E")

- Degree centrality
  - How many connections the node has
- Betweenness centrality
  - How many shortest paths *go* through the node
- Closeness centrality
  - How close the node is to other nodes
- Eigenvector centrality

Degree centrality

- Simply, # of links attached to a node

$$C_D(v) = deg(v)$$

or sometimes defined as

$$C_D(v) = deg(v) / (N-1)$$

Betweenness centrality

- Prob. for a node to be on shortest paths between two other nodes

$$C_B(v) = \frac{1}{(n-1)(n-2)} \sum_{s \neq v, e \neq v} \frac{\#sp_{(s,e,v)}}{\#sp_{(s,e)}}$$

  - s: start node, e: end node
  - $\#sp_{(s,e,v)}$: # of shortest paths from s to e that go though node v
  - $\#sp_{(s,e)}$: total # of shortest paths from s to e
  - Easily generalizable to "group betweenness"

Closeness centrality

- Inverse of an average distance from a node to all the other nodes

$$C_C(v) = \frac{n-1}{\Sigma_{w \neq v}\, d(v,w)}$$

  - d(v,w): length of the shortest path from v to w
  - Its inverse is called "farness"
  - Sometimes "Σ" is moved out of the fraction (it works for networks that are not strongly connected)
  - NetworkX calculates closeness within each connected component

Eigenvector centrality

- Eigenvector of the largest eigenvalue of the adjacency matrix of a network

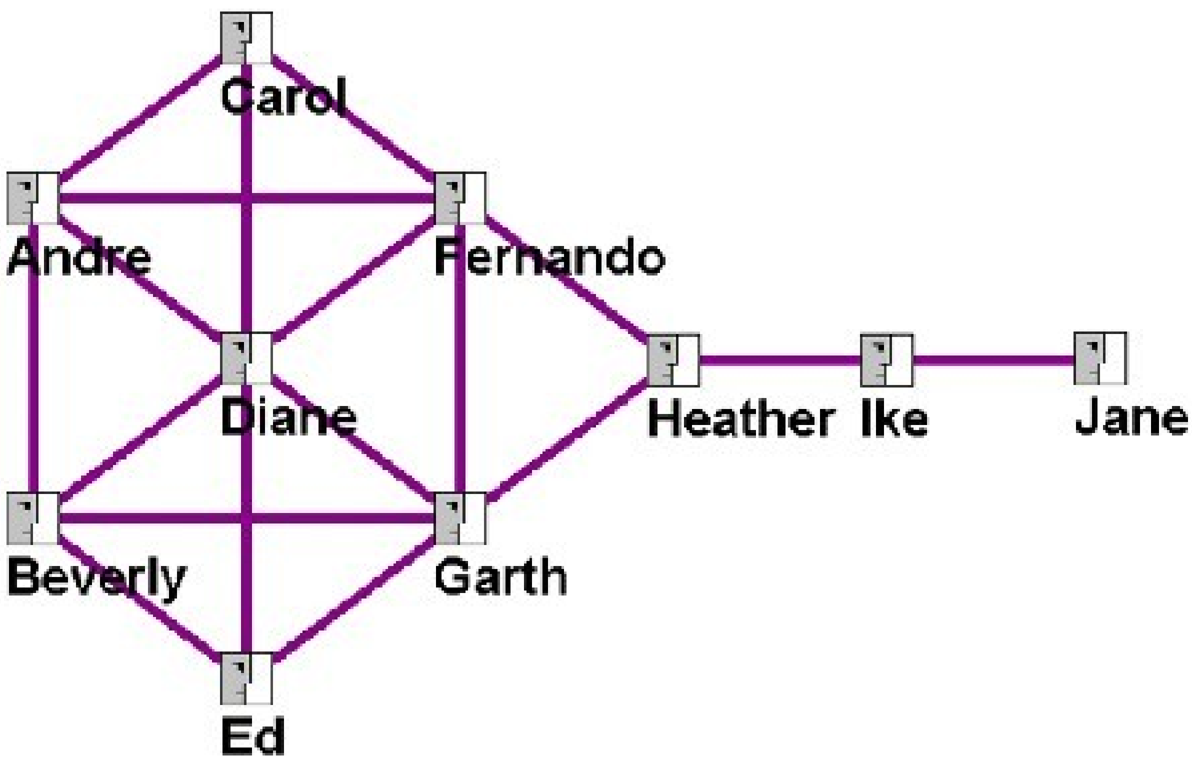
$$C_E(v) = (v\text{-th element of } x)$$

$$Ax = \lambda x$$

- λ: dominant eigenvalue
- x is often normalized ($|x| = 1$)

Exercise

- **Who is most central by degree, betweenness, closeness, eigenvector?**



Which centrality to use?

- To find the most popular person
- To find the most efficient person to collect information from the entire organization
- To find the most powerful person to control information flow within an organization
- To find the *most important* person (?)

Exercise

- Measure four different centralities for all nodes in the Karate Club network and visualize the network by coloring nodes with their centralities

Exercise

- Create a directed network of any kind and measure centralities

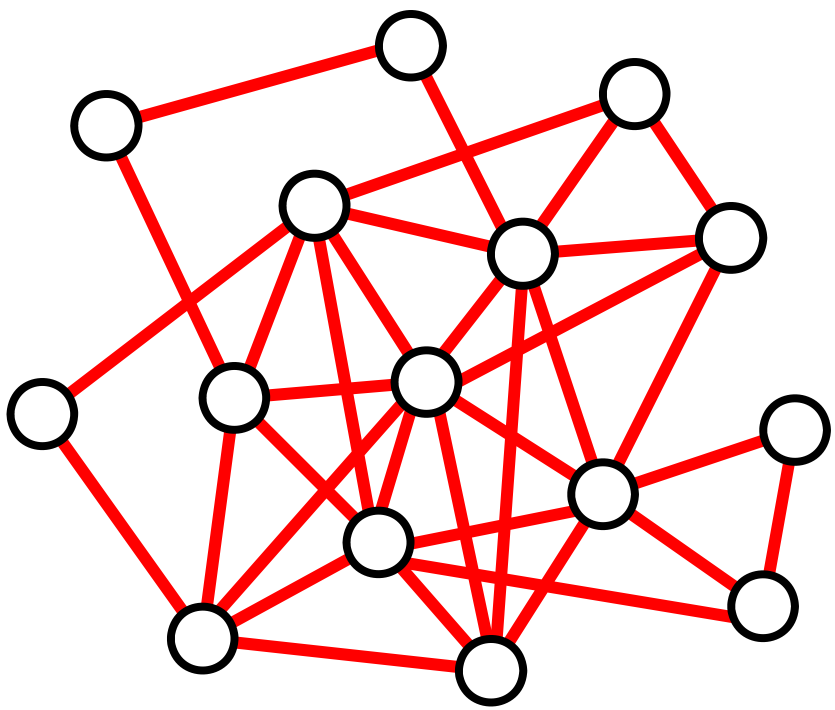
- Make it undirected and do the same
  - How are the centrality measures affected?

K-core

- A connected component of a network obtained by repeatedly deleting all the nodes whose degree is less than k until no more such nodes exist
  - Helps identify where the core cluster is
  - All nodes of a k-core have at least degree k
  - The largest value of k for which a k-core exists is called "degeneracy" of the network

Exercise

- Find the k-core (with the largest k) of the following network



Coreness (core number)

- A node's coreness (core number) is c if it belongs to a c-core but not (c+1)-core

- Indicates how strongly the node is connected to the network
- Classifies nodes into several layers
  - Useful for visualization

Exercise

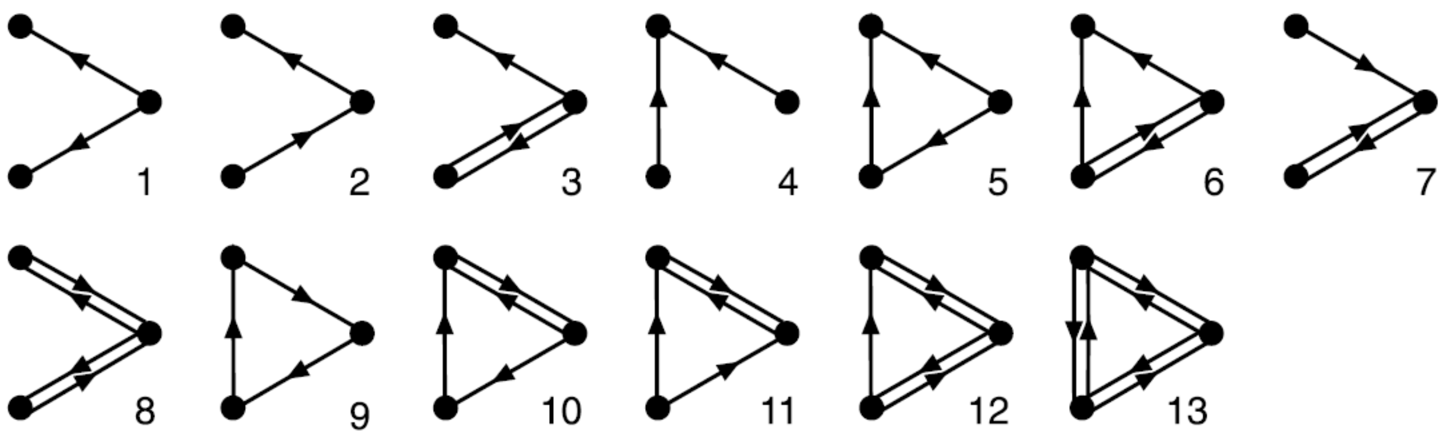
- Obtain the k-core (for largest k) of the Karate Club graph and visualize it
- Calculate the coreness of its nodes and plot its histogram

- Do the same for the (undirected) Supreme Court citation network

Mesoscopic Structures

Motifs

- Small patterns of connections in a network whose number of appearance is significantly higher than those in randomized networks



(from Milo et al., Science 298: 824-827, 2002)

Network	Nodes	Edges	N_{real}	$N_{rand} \pm SD$	Z score	N_{real}	$N_{rand} \pm SD$	Z score	N_{real}	$N_{rand} \pm SD$	Z score
Gene regulation (transcription)					**Feed-forward loop**			**Bi-fan**			
E. coli	424	519	40	7 ± 3	10	203	47 ± 12	13			
*S. cerevisiae**	685	1,052	70	11 ± 4	14	1812	300 ± 40	41			
Neurons					**Feed-forward loop**			**Bi-fan**			**Bi-parallel**
C. elegans†	252	509	125	90 ± 10	3.7	127	55 ± 13	5.3	227	35 ± 10	20
Food webs					**Three chain**			**Bi-parallel**			
Little Rock	92	984	3219	3120 ± 50	2.1	7295	2220 ± 210	25			
Ythan	83	391	1182	1020 ± 20	7.2	1357	230 ± 50	23			
St. Martin	42	205	469	450 ± 10	NS	382	130 ± 20	12			
Chesapeake	31	67	80	82 ± 4	NS	26	5 ± 2	8			
Coachella	29	243	279	235 ± 12	3.6	181	80 ± 20	5			
Skipwith	25	189	184	150 ± 7	5.5	397	80 ± 25	13			
B. Brook	25	104	181	130 ± 7	7.4	267	30 ± 7	32			
Electronic circuits (forward logic chips)					**Feed-forward loop**			**Bi-fan**			**Bi-parallel**
s15850	10,383	14,240	424	2 ± 2	285	1040	1 ± 1	1200	480	2 ± 1	335
s38584	20,717	34,204	413	10 ± 3	120	1739	6 ± 2	800	711	9 ± 2	320
s38417	23,843	33,661	612	3 ± 2	400	2404	1 ± 1	2550	531	2 ± 2	340
s9234	5,844	8,197	211	2 ± 1	140	754	1 ± 1	1050	209	1 ± 1	200
s13207	8,651	11,831	403	2 ± 1	225	4445	1 ± 1	4950	264	2 ± 1	200
Electronic circuits (digital fractional multipliers)					**Three-node feedback loop**			**Bi-fan**			**Four-node feedback loop**
s208	122	189	10	1 ± 1	9	4	1 ± 1	3.8	5	1 ± 1	5
s420	252	399	20	1 ± 1	18	10	1 ± 1	10	11	1 ± 1	11
s838‡	512	819	40	1 ± 1	38	22	1 ± 1	20	23	1 ± 1	25
World Wide Web					**Feedback with two mutual dyads**			**Fully connected triad**			**Uplinked mutual dyad**
nd.edu§	325,729	1.46e6	1.1e5	2e3 ± 1e2	800	6.8e6	5e4±4e2	15,000	1.2e6	1e4 ± 2e2	5000

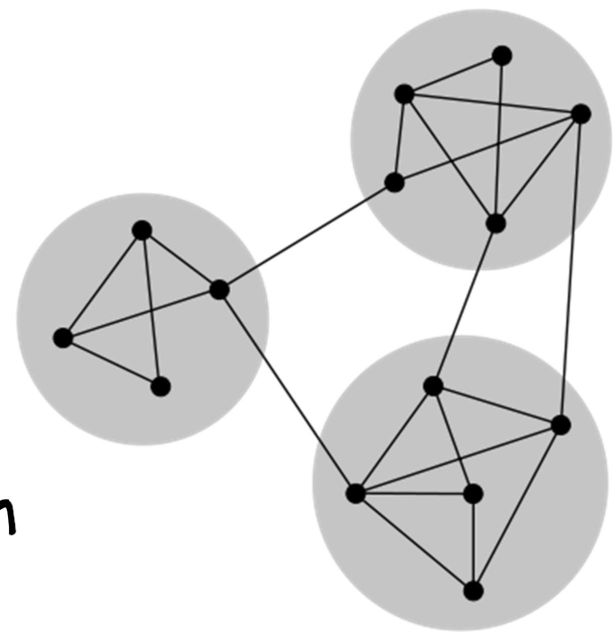
(from Milo et al., Science 298: 824-827, 2002)

Unfortunately…

- Motif counting is computationally costly and still being actively studied, so NetworkX does not have built-in motif counting tools
- One should use specialized software
  - “mfinder” developed at Weizmann Institute of Science
  - “iGraph” in R / Python also has motif counting functions

Community

- A subgraph of a network within which nodes are connected to each other more densely than to the outside
  - Still defined vaguely…
  - Various detection algorithms proposed
    - K-clique percolation
    - Hierarchical clustering
    - Girvan-Newman algorithm
    - Modularity maximization (e.g., Louvain method)



(diagram from Wikipedia)

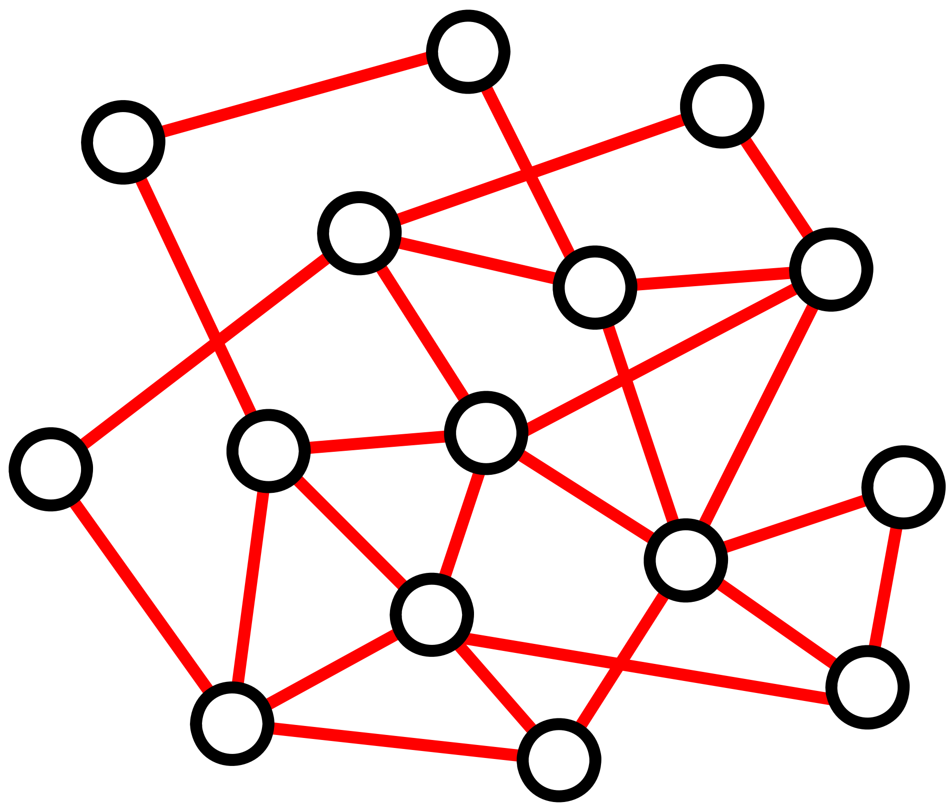
K-clique percolation method

1. Choose a value for k (e.g., 4)
2. Find all k-cliques (complete subgraphs of k-nodes) in the network
3. Assume that two cliques belong to the same community if they share k-1 nodes ("k-clique percolation")

- This methods detect communities that potentially overlap

Exercise

- Find communities in the following network by 3-clique percolation



Exercise

- Generate a random network made of 100 nodes and 250 links
- Calculate node positions using spring layout
- Visualize the original network & its k-clique communities (for k = 3 or 4) using the same positions

Exercise

- Find k-clique communities in the (undirected) Supreme Court Citation Network

- Start with large k (say 100) and decrease it until you find a meaningful community

Non-overlapping communities

- Other methods find ways to assign ALL the nodes to one and only one community
  - Community structure is a mapping from a node ID to a community ID
  - No community overlaps
  - No "stray" nodes

Modularity

- A quantity that characterizes how good a given community structure is in dividing the network

$$Q = \frac{|E_{in}| - |E_{in-R}|}{|E|}$$

- $|E_{in}|$: # of links connecting nodes that belong to the same community
- $|E_{in-R}|$: Estimated $|E_{in}|$ if links were random

Community detection based on modularity

- The Louvain method
  - Heuristic algorithm to construct communities that optimize modularity
    - Blondel et al. J. Stat. Mech. 2008 (10): P10008
- Python implementation by Thomas Aynaud available at:
  - https://bitbucket.org/taynaud/python-louvain/

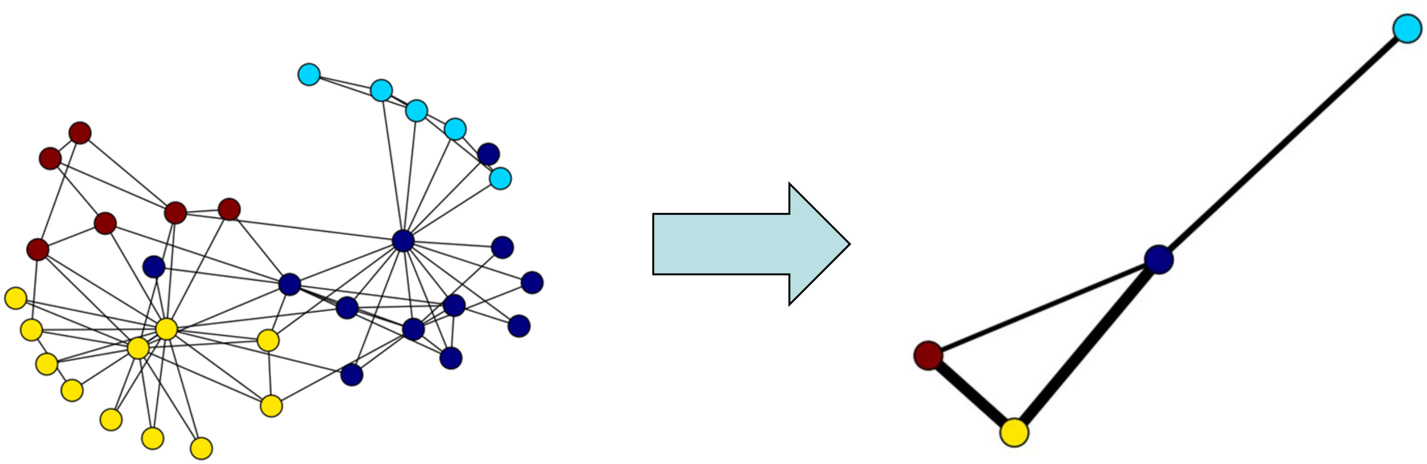
Exercise

- Detect community structure in the (undirected) Supreme Court Citation Network using the Louvain method

- Measure the modularity achieved
- How many communities are detected?
- How large is each community?

Block model

- Create a new, "coarse" network by aggregating nodes within each community into a meta-node
  - Meta-nodes contain original communities
  - Meta-edge weights show connections b/w communities



Exercise

- Create a block model of some real-world network by using its communities as partitions

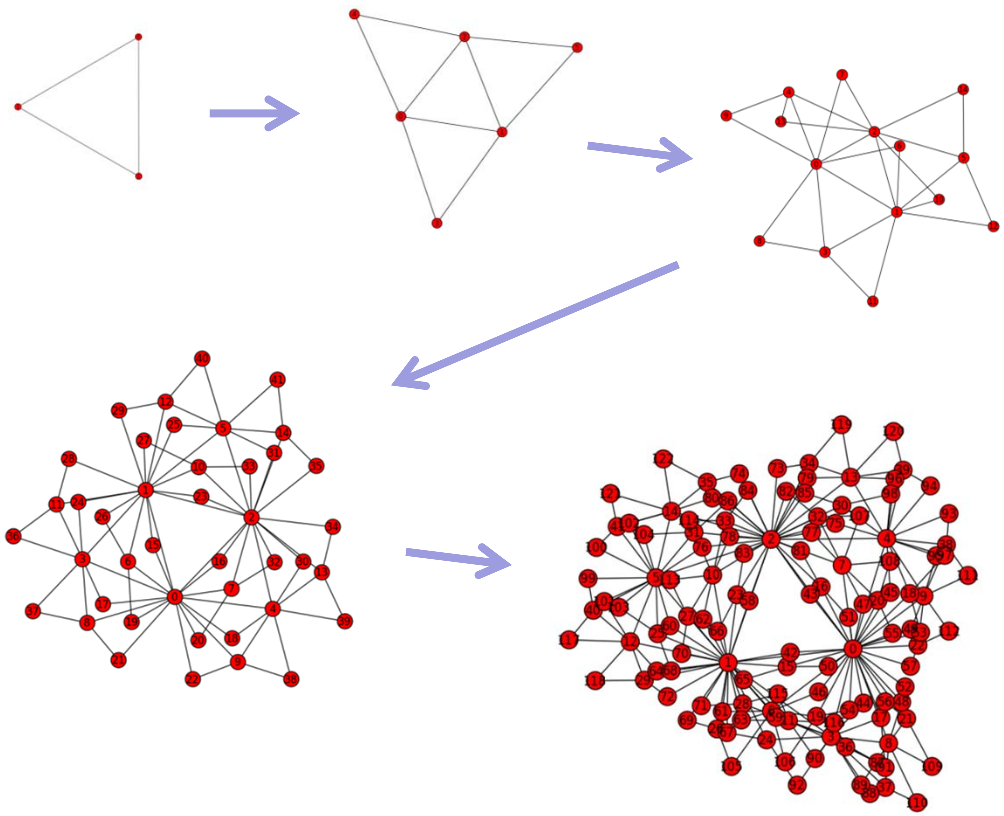
- Visualize the block model with edge widths varied according to connections between communities

Hierarchy

- Many real-world complex networks have many layers of modular structures forming a hierarchy
  - Community structures are not single-scale, but multiscale
  - Similar to fractals

Deterministic scale-free networks

- E.g. Dorogovtsev, Goltsev & Mendes 2002
  - Scale-free degree distribution
  - But still high clustering coefficients

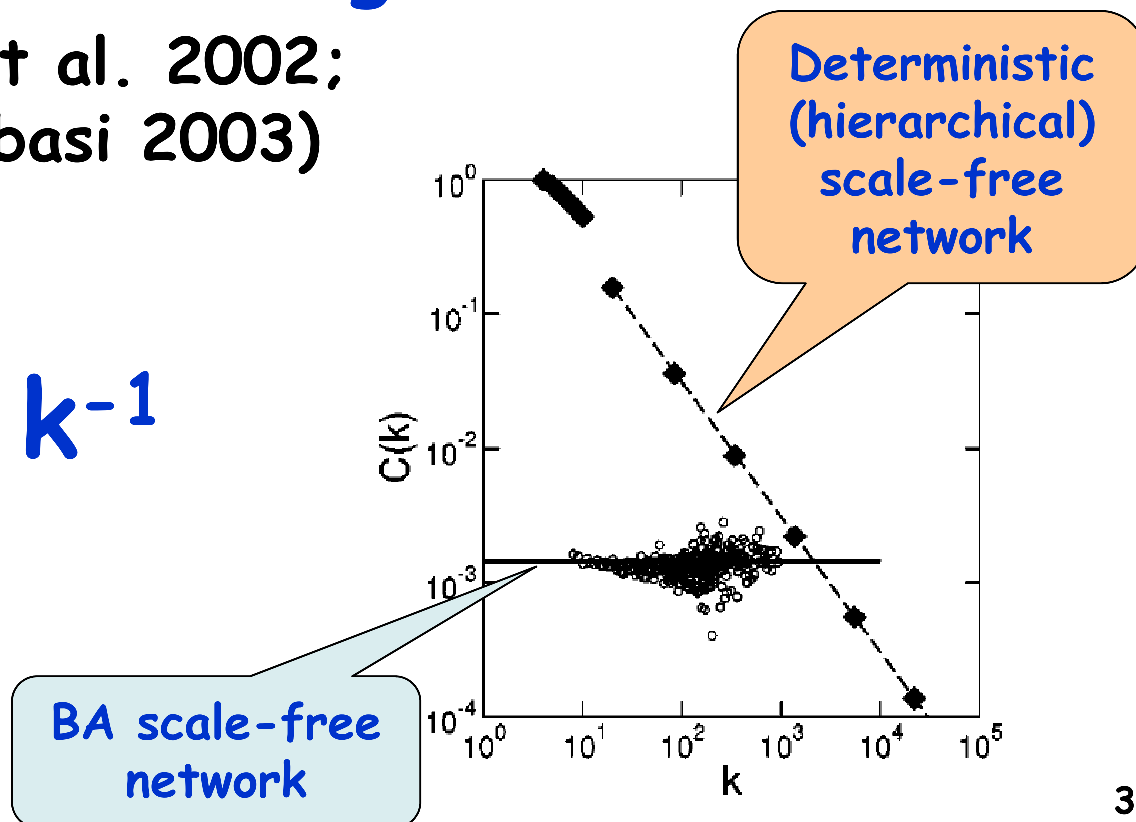


Clustering coefficients and k

- Deterministic scale-free networks show another scaling law

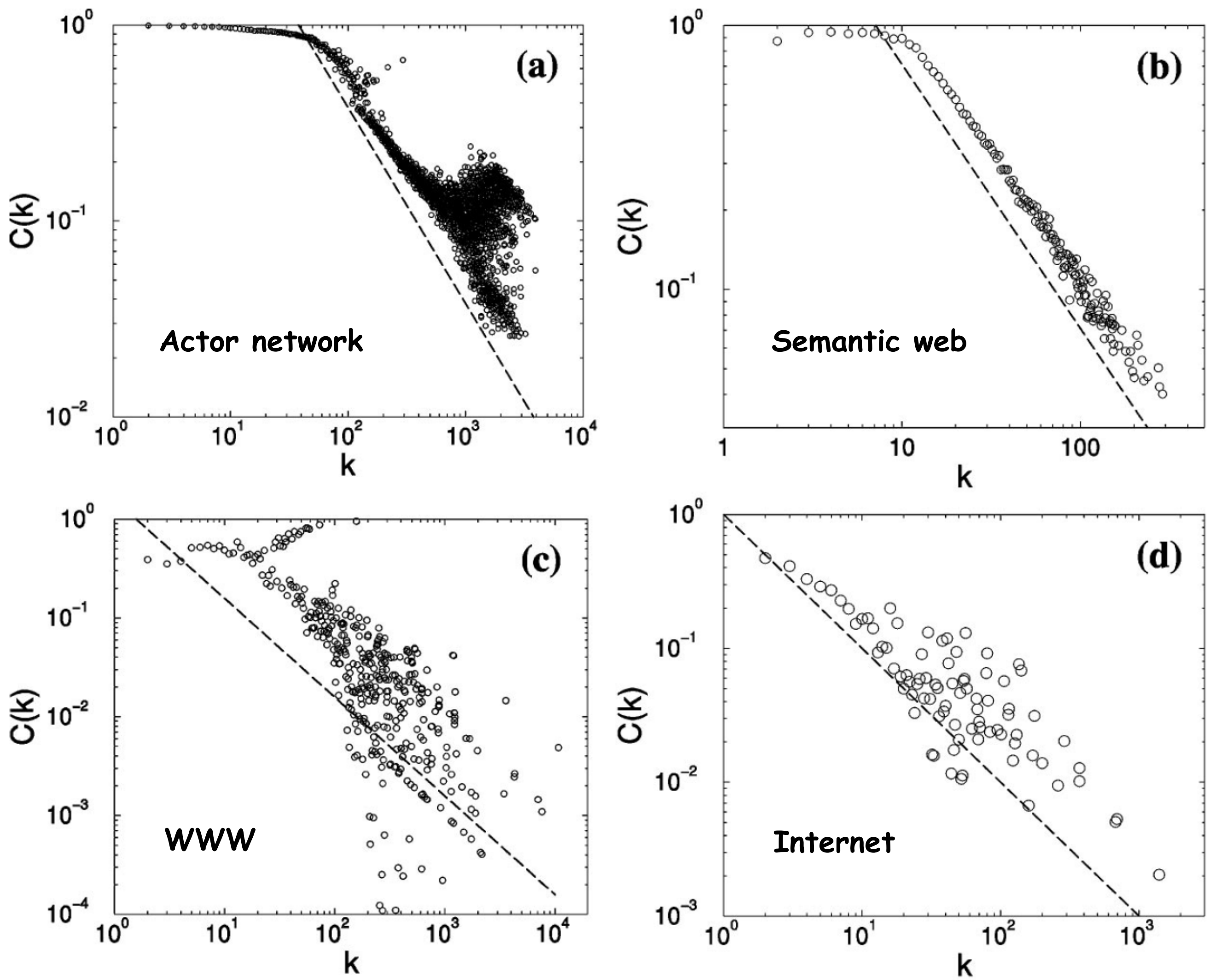
(Dorogovtsev et al. 2002; Ravasz & Barabasi 2003)

$$C(k) \sim k^{-1}$$



(from Ravasz & Barabasi 2003)

C(k) plots of real-world networks



(from Ravasz & Barabasi 2003)

Exercise

- Plot $C(k)$ for several real-world network data and see if the inverse scaling law between k and $C(k)$ appears or not