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

\[ C_D(v) = \text{deg}(v) \]

or sometimes defined as

\[ C_D(v) = \text{deg}(v) / (N-1) \]
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 through 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

\[ C_c(v) = \frac{n-1}{\sum_{w \neq v} d(v, w)} \]

• \(d(v, w)\): length of the shortest path from \(v\) to \(w\)
• Its inverse is called “farness”
• Sometimes “\(\Sigma\)” 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

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

$$Ax = \lambda x$$

- $\lambda$: 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)
<table>
<thead>
<tr>
<th>Network</th>
<th>Nodes</th>
<th>Edges</th>
<th>$N_{\text{real}}$</th>
<th>$N_{\text{rand}}$ ± SD</th>
<th>Z score</th>
<th>$N_{\text{real}}$</th>
<th>$N_{\text{rand}}$ ± SD</th>
<th>Z score</th>
<th>$N_{\text{real}}$</th>
<th>$N_{\text{rand}}$ ± SD</th>
<th>Z score</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gene regulation (transcription)</td>
<td></td>
<td></td>
<td>X \downarrow \quad Y \downarrow \quad Z \downarrow</td>
<td>Feed-forward loop</td>
<td></td>
<td>X \downarrow \quad Z \downarrow \quad W \downarrow</td>
<td>Bi-fan</td>
<td></td>
<td>X \downarrow \quad Y \downarrow \quad Z \downarrow</td>
<td>Bi-fan</td>
<td></td>
</tr>
<tr>
<td>\textit{E. coli}</td>
<td>424</td>
<td>519</td>
<td>40 ± 7</td>
<td>10</td>
<td>203</td>
<td>47 ± 12</td>
<td>13</td>
<td></td>
<td>1812</td>
<td>300 ± 40</td>
<td>41</td>
</tr>
<tr>
<td>\textit{S. cerevisiae}*</td>
<td>685</td>
<td>1,052</td>
<td>70 ± 4</td>
<td>14</td>
<td>1812</td>
<td>300 ± 40</td>
<td>41</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Neurons</td>
<td></td>
<td></td>
<td>X \downarrow \quad Y \downarrow \quad Z \downarrow</td>
<td>Feed-forward loop</td>
<td></td>
<td>X \downarrow \quad Z \downarrow \quad W \downarrow</td>
<td>Bi-fan</td>
<td></td>
<td>X \downarrow \quad Y \downarrow \quad Z \downarrow</td>
<td>Bi-parallel</td>
<td></td>
</tr>
<tr>
<td>\textit{C. elegans}\†</td>
<td>252</td>
<td>509</td>
<td>125 ± 10</td>
<td>3.7</td>
<td>127</td>
<td>55 ± 13</td>
<td>5.3</td>
<td></td>
<td>227</td>
<td>35 ± 10</td>
<td>20</td>
</tr>
<tr>
<td>Food webs</td>
<td></td>
<td></td>
<td>X \downarrow \quad Y \downarrow \quad Z \downarrow</td>
<td>Three chain</td>
<td></td>
<td>X \downarrow \quad Y \downarrow \quad Z \downarrow</td>
<td>Bi-parallel</td>
<td></td>
<td>X \downarrow \quad Y \downarrow \quad Z \downarrow</td>
<td>Bi-parallel</td>
<td></td>
</tr>
<tr>
<td>Little Rock</td>
<td>92</td>
<td>984</td>
<td>3219 ± 3120</td>
<td>2.1</td>
<td>7295</td>
<td>2220 ± 210</td>
<td>25</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ythan</td>
<td>83</td>
<td>391</td>
<td>1182 ± 1020</td>
<td>7.2</td>
<td>1357</td>
<td>230 ± 50</td>
<td>23</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>St. Martin</td>
<td>42</td>
<td>205</td>
<td>469 ± 450</td>
<td>NS</td>
<td>382</td>
<td>130 ± 20</td>
<td>12</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Chesapeake</td>
<td>31</td>
<td>67</td>
<td>80 ± 82</td>
<td>NS</td>
<td>26</td>
<td>5 ± 2</td>
<td>8</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Coachella</td>
<td>29</td>
<td>243</td>
<td>279 ± 235</td>
<td>3.6</td>
<td>181</td>
<td>80 ± 20</td>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Skipwith</td>
<td>25</td>
<td>189</td>
<td>184 ± 150</td>
<td>5.5</td>
<td>397</td>
<td>80 ± 25</td>
<td>13</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>B. Brook</td>
<td>25</td>
<td>104</td>
<td>181 ± 130</td>
<td>7.4</td>
<td>267</td>
<td>30 ± 7</td>
<td>32</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Electronic circuits</td>
<td></td>
<td></td>
<td>X \downarrow \quad Y \downarrow \quad Z \downarrow</td>
<td>Feed-forward loop</td>
<td></td>
<td>X \downarrow \quad Z \downarrow \quad W \downarrow</td>
<td>Bi-fan</td>
<td></td>
<td>X \downarrow \quad Y \downarrow \quad Z \downarrow</td>
<td>Bi-parallel</td>
<td></td>
</tr>
<tr>
<td>(forward logic chips)</td>
<td></td>
<td></td>
<td>s15850 31424</td>
<td>2 ± 2</td>
<td>1040</td>
<td>1 ± 1</td>
<td>1200</td>
<td></td>
<td>480</td>
<td>2 ± 1</td>
<td>335</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>s38584 32045</td>
<td>10 ± 3</td>
<td>1739</td>
<td>6 ± 2</td>
<td>800</td>
<td></td>
<td>711</td>
<td>9 ± 2</td>
<td>320</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>s38417 33661</td>
<td>2 ± 2</td>
<td>2340</td>
<td>1 ± 1</td>
<td>2550</td>
<td></td>
<td>513</td>
<td>2 ± 2</td>
<td>340</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>s9234 18917</td>
<td>2 ± 1</td>
<td>211</td>
<td>1 ± 1</td>
<td>1950</td>
<td></td>
<td>209</td>
<td>1 ± 1</td>
<td>200</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>s13207 11831</td>
<td>1 ± 1</td>
<td>403</td>
<td>2 ± 1</td>
<td>4950</td>
<td></td>
<td>264</td>
<td>2 ± 1</td>
<td>200</td>
</tr>
<tr>
<td>Electronic circuits</td>
<td></td>
<td></td>
<td>X \downarrow \quad Y \downarrow \quad Z \downarrow</td>
<td>Three-node feedback loop</td>
<td></td>
<td>X \downarrow \quad Y \downarrow \quad Z \downarrow</td>
<td>Bi-fan</td>
<td></td>
<td>X \downarrow \quad Y \downarrow \quad Z \downarrow</td>
<td>Four-node feedback loop</td>
<td></td>
</tr>
<tr>
<td>(digital fractional multipliers)</td>
<td></td>
<td></td>
<td>s208 122189</td>
<td>1 ± 1</td>
<td>10</td>
<td>1 ± 1</td>
<td>9</td>
<td></td>
<td>4</td>
<td>1 ± 1</td>
<td>3.8</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>s420 252399</td>
<td>1 ± 1</td>
<td>20</td>
<td>1 ± 1</td>
<td>18</td>
<td></td>
<td>10</td>
<td>1 ± 1</td>
<td>10</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>s838\‡ 512819</td>
<td>1 ± 1</td>
<td>40</td>
<td>1 ± 1</td>
<td>38</td>
<td></td>
<td>22</td>
<td>1 ± 1</td>
<td>20</td>
</tr>
<tr>
<td>World Wide Web</td>
<td></td>
<td></td>
<td>X \downarrow \quad Y \downarrow \quad Z \downarrow</td>
<td>Feedback with two mutual dyads</td>
<td></td>
<td>X \downarrow \quad Y \downarrow \quad Z \downarrow</td>
<td>Fully connected triad</td>
<td></td>
<td>X \downarrow \quad Y \downarrow \quad Z \downarrow</td>
<td>Uplinked mutual dyad</td>
<td></td>
</tr>
<tr>
<td>nd.edu§</td>
<td>325,729</td>
<td>1.46e6</td>
<td>1.1e5 2e3 ± 1e2</td>
<td>800</td>
<td>6.8e6 5e4±4e2</td>
<td>15,000</td>
<td>1.2e6 1e4 ± 2e2</td>
<td>5000</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

(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)
**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 method detects 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

• 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