Motifs and Graphlets

CS224W: Analysis of Networks
Jure Leskovec, Stanford University
http://cs224w.stanford.edu
Network Metrics

- Many metrics at the node level:
  - E.g., node degree, PageRank score, node clustering
- Many metrics at the whole-network level:
  - E.g., diameter, clustering, size of giant component
- What about something in-between?
  - A mesoscale characterization of networks
Subnetworks, or subgraphs, are the building blocks of networks:

- They have the power to **characterize** and **discriminate** networks
Building Blocks of Networks

Subgraph decomposition of an electronic circuit
Let’s consider all possible (non-isomorphic) directed subgraphs of size 3
For each subgraph:
- Imagine you have a metric capable of classifying the subgraph “significance” [more on that later]
  - Negative values indicate under-representation
  - Positive values indicate over-representation

We create a network significance profile:
- A feature vector with values for all subgraph types

Next: Compare profiles of different networks:
- Regulatory network (gene regulation)
- Neuronal network (synaptic connections)
- World Wide Web (hyperlinks between pages)
- Social network (friendships)
- Language networks (word adjacency)
Different networks have similar significance profiles!

Web and social
Gene regulation networks
Neurons
Language networks

Network significance profile

Milo et al., Science 2004

Jure Leskovec, Stanford CS224W: Analysis of Networks
Clustering of networks based on their significance profiles

Correlation in significance profile of the English and French language networks

Closely related networks have more similar significance profiles

Milo et al., Science 2004
Co-authorship Network Example

Subgraph types (corresponding to the X-axis of the plot)

Network significance profile

Networks
- Computer Science
- Engineering
- Environment Ecology
- Materials Science
- Pharmacology Toxicology

Choobdar et al., ASONAM 2012
Subgraphs, Motifs, and Graphlets
Network Motifs

- **Network motifs**: “recurring, significant patterns of interconnections”

- **How to define a network motif:**
  - **Pattern**: induced/non-induced subgraph
  - **Recurring**: found many times, i.e., with high frequency
  - **Significant**: more frequent than expected, i.e., in randomly generated networks
    - Erdos-Renyi random graphs, scale-free networks
Why Do We Need Motifs?

- **Motifs:**
  - Help us understand how networks work
  - Help us predict operation and reaction of the network in a given situation

- **Examples:**
  - **Feed-forward loops:** found in networks of neurons, where they neutralize “biological noise”
  - **Parallel loops:** found in food webs
  - **Single-input modules:** found in gene control networks
Induced subgraph of interest (aka Motif):

No match!

Match!
Motifs: Recurrence

Motif of interest:

- Allow **overlapping of motifs**
- Network on the right has 4 occurrences of the motif:
  - \{1,2,3,4,5\}
  - \{1,2,3,4,6\}
  - \{1,2,3,4,7\}
  - \{1,2,3,4,8\}
**Significance of a Motif**

- **Key idea:** Subgraphs that occur in a real network much more often than in a random network have functional significance.

---

**Milo et. al., Science 2002**
Significance of a Motif

- Motifs are **overrepresented** in a network when compared to **randomized networks**:
  - $Z_i$ captures **statistical significance of motif** $i$:
    \[ Z_i = \frac{(N_i^{\text{real}} - N_i^{\text{rand}})}{\text{std}(N_i^{\text{rand}})} \]
    - $N_i^{\text{real}}$ is #(subgraphs of type $i$) in network $G^{\text{real}}$
    - $N_i^{\text{rand}}$ is #(subgraphs of type $i$) in randomized network $G^{\text{rand}}$

- **Network significance profile**:
  \[ SP_i = Z_i / \sqrt{\sum Z_j^2} \]
  - $SP$ is a vector of **normalized Z-scores**
  - $SP$ emphasizes relative significance of subgraphs:
    - Important for comparison of networks of different sizes
    - Generally, larger networks display higher Z-scores
Goal: Generate a random graph with a given degree sequence $k_1, k_2, \ldots, k_N$

Useful as a “null” model of networks:
- We can compare the real network $G$ and a “random” $G'$ which has the same degree sequence as $G$

Configuration model:

Nodes with spokes

Randomly pair up “mini”-nodes

Resulting graph

We ignore double edges and self-loops when creating the final graph
Alternative for Spokes: **Switching**

- Start from a **given graph** $G$
- Repeat the **switching step** $Q|E(G)|$ times:
  - Select a pair of edges $A \rightarrow B$, $C \rightarrow D$ at random
  - **Exchange** the endpoints to give $A \rightarrow D$, $C \rightarrow B$
    - Exchange edges only if no multiple edges or self-edges are generated

- **Result:** A randomly rewired graph:
  - Same node degrees, randomly rewired edges

- $Q$ is chosen large enough (e.g., $Q = 100$) for the process to converge
Motifs: Significance Example

Network significance profile

\[ Z_i = \frac{(N_{i}^{\text{real}} - \bar{N}_{i}^{\text{rand}})}{\text{std}(N_{i}^{\text{rand}})} \]
RECAP: Detecting Motifs

- Count subgraphs $i$ in $G^{\text{real}}$
- Count subgraphs $i$ in random networks $G^{\text{rand}}$:
  - **Configuration model**: Each $G^{\text{rand}}$ has the same
    #nodes, #edges and #degree distribution as $G^{\text{real}}$

- Assign **Z-score** to $i$:
  - $Z_i = (N_i^{\text{real}} - \bar{N}_i^{\text{rand}})/\text{std}(N_i^{\text{rand}})$
  - **High Z-score**: Subgraph $i$
    is a network motif of $G$
Variations on the Motif Concept

- **Canonical definition:**
  - Directed and undirected
  - Colored and uncolored
  - Temporal and static motifs

- **Variations on the concept**
  - Different frequency concepts
  - Different significance metrics
  - Under-Representation (anti-motifs)
  - Different constraints for null model
## Experiments: Detecting Motifs

<table>
<thead>
<tr>
<th>Network</th>
<th>Nodes</th>
<th>Edges</th>
<th>$N_{\text{real}}$</th>
<th>$N_{\text{rand}} \pm SD$</th>
<th>$Z$ score</th>
<th>$N_{\text{real}}$</th>
<th>$N_{\text{rand}} \pm SD$</th>
<th>$Z$ score</th>
<th>$N_{\text{real}}$</th>
<th>$N_{\text{rand}} \pm SD$</th>
<th>$Z$ score</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Gene regulation</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(transcription)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><em>E. coli</em></td>
<td>424</td>
<td>519</td>
<td>40</td>
<td>7 ± 3</td>
<td>10</td>
<td>203</td>
<td>47 ± 12</td>
<td>13</td>
<td>1812</td>
<td>300 ± 40</td>
<td>41</td>
</tr>
<tr>
<td><em>S. cerevisiae</em>†</td>
<td>685</td>
<td>1,052</td>
<td>70</td>
<td>11 ± 4</td>
<td>14</td>
<td>1812</td>
<td>300 ± 40</td>
<td>41</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Neurons</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>C. elegans†</strong></td>
<td>252</td>
<td>509</td>
<td>125</td>
<td>90 ± 10</td>
<td>3.7</td>
<td>127</td>
<td>55 ± 13</td>
<td>5.3</td>
<td>227</td>
<td>35 ± 10</td>
<td>20</td>
</tr>
<tr>
<td><strong>Food webs</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Little Rock</td>
<td>92</td>
<td>984</td>
<td>3219</td>
<td>3120 ± 50</td>
<td>2.1</td>
<td>7295</td>
<td>2220 ± 210</td>
<td>25</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ythan</td>
<td>83</td>
<td>391</td>
<td>1182</td>
<td>1020 ± 20</td>
<td>7.2</td>
<td>1357</td>
<td>230 ± 50</td>
<td>23</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>St. Martin</td>
<td>42</td>
<td>205</td>
<td>469</td>
<td>450 ± 10</td>
<td>NS</td>
<td>382</td>
<td>130 ± 20</td>
<td>12</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Chesapeake</td>
<td>31</td>
<td>67</td>
<td>80</td>
<td>82 ± 4</td>
<td>NS</td>
<td>26</td>
<td>5 ± 2</td>
<td>8</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Coachella</td>
<td>29</td>
<td>243</td>
<td>279</td>
<td>235 ± 12</td>
<td>3.6</td>
<td>181</td>
<td>80 ± 20</td>
<td>5</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Skipwith</td>
<td>25</td>
<td>189</td>
<td>184</td>
<td>150 ± 7</td>
<td>5.5</td>
<td>397</td>
<td>80 ± 25</td>
<td>13</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>B. Brook</td>
<td>25</td>
<td>104</td>
<td>181</td>
<td>130 ± 7</td>
<td>7.4</td>
<td>267</td>
<td>30 ± 7</td>
<td>32</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Z-scores of individual motifs for different networks
# Experiments: Detecting Motifs

<table>
<thead>
<tr>
<th>Network</th>
<th>Nodes</th>
<th>Edges</th>
<th>$N_{\text{real}}$</th>
<th>$N_{\text{rand}} \pm \text{SD}$</th>
<th>$Z$ score</th>
<th>$N_{\text{real}}$</th>
<th>$N_{\text{rand}} \pm \text{SD}$</th>
<th>$Z$ score</th>
<th>$N_{\text{real}}$</th>
<th>$N_{\text{rand}} \pm \text{SD}$</th>
<th>$Z$ score</th>
</tr>
</thead>
<tbody>
<tr>
<td>Electronic circuits (forward logic chips)</td>
<td>s15850</td>
<td>10,383</td>
<td>14,240</td>
<td>424</td>
<td>2 ± 2</td>
<td>285</td>
<td>1040</td>
<td>1 ± 1</td>
<td>1200</td>
<td>480</td>
<td>2 ± 1</td>
</tr>
<tr>
<td></td>
<td>s38584</td>
<td>20,717</td>
<td>34,204</td>
<td>413</td>
<td>10 ± 3</td>
<td>120</td>
<td>1739</td>
<td>6 ± 2</td>
<td>800</td>
<td>711</td>
<td>9 ± 2</td>
</tr>
<tr>
<td></td>
<td>s38417</td>
<td>23,843</td>
<td>33,661</td>
<td>612</td>
<td>3 ± 2</td>
<td>400</td>
<td>2404</td>
<td>1 ± 1</td>
<td>2550</td>
<td>531</td>
<td>2 ± 2</td>
</tr>
<tr>
<td></td>
<td>s9234</td>
<td>5,844</td>
<td>8,197</td>
<td>211</td>
<td>2 ± 1</td>
<td>140</td>
<td>754</td>
<td>1 ± 1</td>
<td>1050</td>
<td>209</td>
<td>1 ± 1</td>
</tr>
<tr>
<td></td>
<td>s13207</td>
<td>8,651</td>
<td>11,831</td>
<td>403</td>
<td>2 ± 1</td>
<td>225</td>
<td>4445</td>
<td>1 ± 1</td>
<td>4950</td>
<td>264</td>
<td>2 ± 1</td>
</tr>
<tr>
<td>Electronic circuits (digital fractional multipliers)</td>
<td>s208</td>
<td>122</td>
<td>189</td>
<td>10</td>
<td>1 ± 1</td>
<td>9</td>
<td>4</td>
<td>1 ± 1</td>
<td>3.8</td>
<td>5</td>
<td>1 ± 1</td>
</tr>
<tr>
<td></td>
<td>s420</td>
<td>252</td>
<td>399</td>
<td>20</td>
<td>1 ± 1</td>
<td>18</td>
<td>10</td>
<td>1 ± 1</td>
<td>10</td>
<td>11</td>
<td>1 ± 1</td>
</tr>
<tr>
<td></td>
<td>s838‡</td>
<td>512</td>
<td>819</td>
<td>40</td>
<td>1 ± 1</td>
<td>38</td>
<td>22</td>
<td>1 ± 1</td>
<td>20</td>
<td>23</td>
<td>1 ± 1</td>
</tr>
<tr>
<td>World Wide Web</td>
<td>nd.edu§</td>
<td>325,729</td>
<td>1.46e6</td>
<td>1.1e5</td>
<td>2e3 ± 1e2</td>
<td>800</td>
<td>6.8e6</td>
<td>5e4±4e2</td>
<td>15,000</td>
<td>1.2e6</td>
<td>1e4 ± 2e2</td>
</tr>
</tbody>
</table>

Z-scores of individual motifs for different networks
What do we learn from prior 2 slides?

- **Network of neurons and a gene network** contain similar motifs:
  - Feed-forward loops and bi-fan structures
  - Both are information processing networks with sensory and acting components

- **Food webs** have parallel loops:
  - Prey of a particular predator share prey

- **WWW network** has bidirectional links
  - Design that allows the shortest path among sets of related pages
**New Concept: Graphlets**

- **Graphlets**: connected non-isomorphic subgraphs
  
  - **Induced subgraphs of any frequency**

<table>
<thead>
<tr>
<th>2-node graphlet</th>
<th>3-node graphlets</th>
<th>4-node graphlets</th>
<th>5-node graphlets</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

For $n = 3, 4, 5, \ldots$ 10 there are 2, 6, 21, \ldots 11716571 graphlets!

---

Przulj et al., Bioinformatics 2004
Next: Use graphlets to obtain a node-level subgraph metric

- **Degree** counts #(edges) that a node touches:
  - Can we generalize this notion for graphlets? – Yes!

- **Graphlet degree vector** counts #(graphlets) that a node touches
An **automorphism orbit** takes into account the symmetries of a subgraph

**Graphlet Degree Vector (GDV):** a vector with the frequency of the node in each orbit position

**Example:** Graphlet degree vector of node $v$

For a node $u$ of graph $G$, the automorphism orbit of $u$ is $\text{Orb}(u) = \{v \in V(G); v = f(u) \text{ for some } f \in \text{Aut}(G)\}$.

The Aut denotes an automorphism group of $G$, i.e., an isomorphism from $G$ to itself.
Graphlet degree vector counts \( \#(\text{graphlets}) \) that a node touches at a particular orbit.

Considering graphlets on 2 to 5 nodes we get:

- Vector of 73 coordinates is a signature of a node that describes the topology of node's neighborhood.
- Captures its interconnectivities out to a distance of 4 hops.

Graphlet degree vector provides a measure of a node’s local network topology:

- Comparing vectors of two nodes provides a highly constraining measure of local topological similarity between them.
Graphlet Degree Vector (GDV) of node A:

- $i$-th element of GDV(A): \#(graphlets) that touch A at orbit $i$
- Highlighted are graphlets that touch node A at orbits 15, 19, 27, and 35 from left to right
Computational Challenge

- Finding size-k motifs/graphlets requires solving two challenges:
  - 1) **Enumerating** all size-k connected subgraphs
  - 2) **Counting** #(occurrences of each subgraph type)

- Just knowing if a certain subgraph exists in a graph is a **hard computational problem**!
  - Subgraph isomorphism is NP-complete

- Computation time grows exponentially as the size of the motif/graphlet increases
  - Feasible motif size is usually small (3 to 8)
Counting Subgraphs

- **Network-centric approaches:**
  - **Step 1)** Enumerate all \textit{k}-connected sets of nodes
  - **Step 2)** Count subgraphs of each type via graph isomorphisms test

- **Algorithms:**
  - Exact subgraph enumeration (ESU) [Wernicke 2006]
  - Kavosh [Kashani et al. 2009]
  - Subgraph sampling [Kashtan et al. 2004]

- **Today:** ESU algorithm
Exact Subgraph Enumeration (ESU)

- **Two sets:**
  - $V_{\text{subgraph}}$: currently constructed subgraph (motif)
  - $V_{\text{extension}}$: set of candidate nodes to extend the motif

- **Idea:** Starting with a node $v$, add those nodes to $V_{\text{extension}}$ set that have two properties:
  - Their node_id must be larger than that of $v$
  - They may only be neighbored to the newly added node $w$ but not to a node already in $V_{\text{subgraph}}$

- ESU is implemented as a **recursive function:**
  - The running of this function can be displayed as a **tree-like structure of depth $k$**, called the **ESU-Tree**
Exact Subgraph Enumeration (ESU)

**Algorithm:** EnumerateSubgraphs($G, k$) (ESU)

**Input:** A graph $G = (V, E)$ and an integer $1 \leq k \leq |V|$.

**Output:** All size-$k$ subgraphs in $G$.

1. for each vertex $v \in V$ do
2. $V_{\text{Extension}} \leftarrow \{u \in N(\{v\}) : u > v\}$
3. call ExtendSubgraph($\{v\}, V_{\text{Extension}}, v$)
4. return

**ExtendSubgraph($V_{\text{Subgraph}}, V_{\text{Extension}}, v$)**

- **E1** if $|V_{\text{Subgraph}}| = k$ then output $G[V_{\text{Subgraph}}]$ and return
- **E2** while $V_{\text{Extension}} \neq \emptyset$ do
- **E3** Remove an arbitrarily chosen vertex $w$ from $V_{\text{Extension}}$
- **E4** $V_{\text{Extension}}' \leftarrow V_{\text{Extension}} \cup \{u \in N_{\text{excl}}(w, V_{\text{Subgraph}}) : u > v\}$
- **E5** call ExtendSubgraph($V_{\text{Subgraph}} \cup \{w\}, V_{\text{Extension}}', v$)
- **E6** return
Nodes in the ESU-tree include two adjoining sets:

- $V_{\text{subgraph}}$: A set of adjacent nodes
- $V_{\text{extension}}$: Nodes adjacent to at least one of the SUB nodes whose numerical labels are larger than the SUB nodes labels

Leaves represent all size-3 induced sub-graphs
So far, we enumerated all size-k subgraphs in the input graph.

Next step of ESU: Classify subgraphs placed in the ESU-Tree leaves into non-isomorphic size-k classes:

- Determine which subgraphs in ESU-Tree leaves are topologically equivalent (isomorphic) and group them into subgraph classes accordingly.
- Use McKay’s nauty algorithm [McKay 1981]
Graph Isomorphism

- Graphs $G$ and $H$ are **isomorphic** if there exists a bijection $f : V(G) \rightarrow V(H)$ such that:
  - Any two nodes $u$ and $v$ of $G$ are adjacent in $G$ iff $f(u)$ and $f(v)$ are adjacent in $H$

- Example: Are $G$ and $H$ topologically equivalent?

```
\begin{array}{|c|c|}
\hline
A & 4 \\
B & 7 \\
C & 1 \\
D & 3 \\
E & 5 \\
F & 8 \\
G & 2 \\
H & 9 \\
I & 6 \\
\hline
\end{array}
```

$G$ and $H$ are isomorphic!

Need to check $9!$ possible bijections between node sets

**Hard computational problem!**
Information About the Course Project
Project is a substantial part of the class
- Students put significant effort, and great results have been obtained in the past

Types of projects:
- (1) Analysis of an interesting dataset with the goal to develop a (new) model or an algorithm
- (2) A test of a model or algorithm (that you have read about or your own) on real & simulated data.
  - Fast algorithms for big graphs. Can be integrated into SNAP.

Other points:
- The project should contain some mathematical analysis, and some experimentation on real or synthetic data
- The result of the project will typically be an 8 page paper, describing the approach, the results, and related work.
- Come to us if you need help with a project idea!
Project proposal: 3-5 pages, teams of up to 3 students

- Project proposal has 3 parts:
  - (0) Quick 200 word abstract
  - (1) Related work / Reaction paper (2-3 pages):
    - Read 3 papers related to the project/class
    - Do reading beyond what was covered in class
    - Think beyond what you read. Don’t take other’s work for granted!
    - 2-3 pages: Summary (~1 page), Critique (~1 page)
  - (2) Proposal (1-2 pages):
    - Clearly define the problem you are solving.
    - How does it relate to what you read for the Reaction paper?
    - What data will you use? (make sure you already have it!)
    - Which algorithm/model will you use/develop? Be specific!
    - How will you evaluate/test your method?

See [http://cs224w.stanford.edu/info.html](http://cs224w.stanford.edu/info.html) for detailed instructions and examples of previous proposals
Announcement: Project Proposal

- **Logistics:**
  1. Register your group on this Google Form: https://goo.gl/forms/2wvoXXLm8R91y4jV2
  2. Submit PDF on GradeScope AND at http://snap.stanford.edu/submit/
  Due in 9 days: Thu Oct 18 at 23:59 PST!

- If you need help/ideas/advice come to Office Hours (Jure and Michele) or email us
Here you can find a list of interesting datasets for your CS224W project:

http://cs224w.stanford.edu/data.html