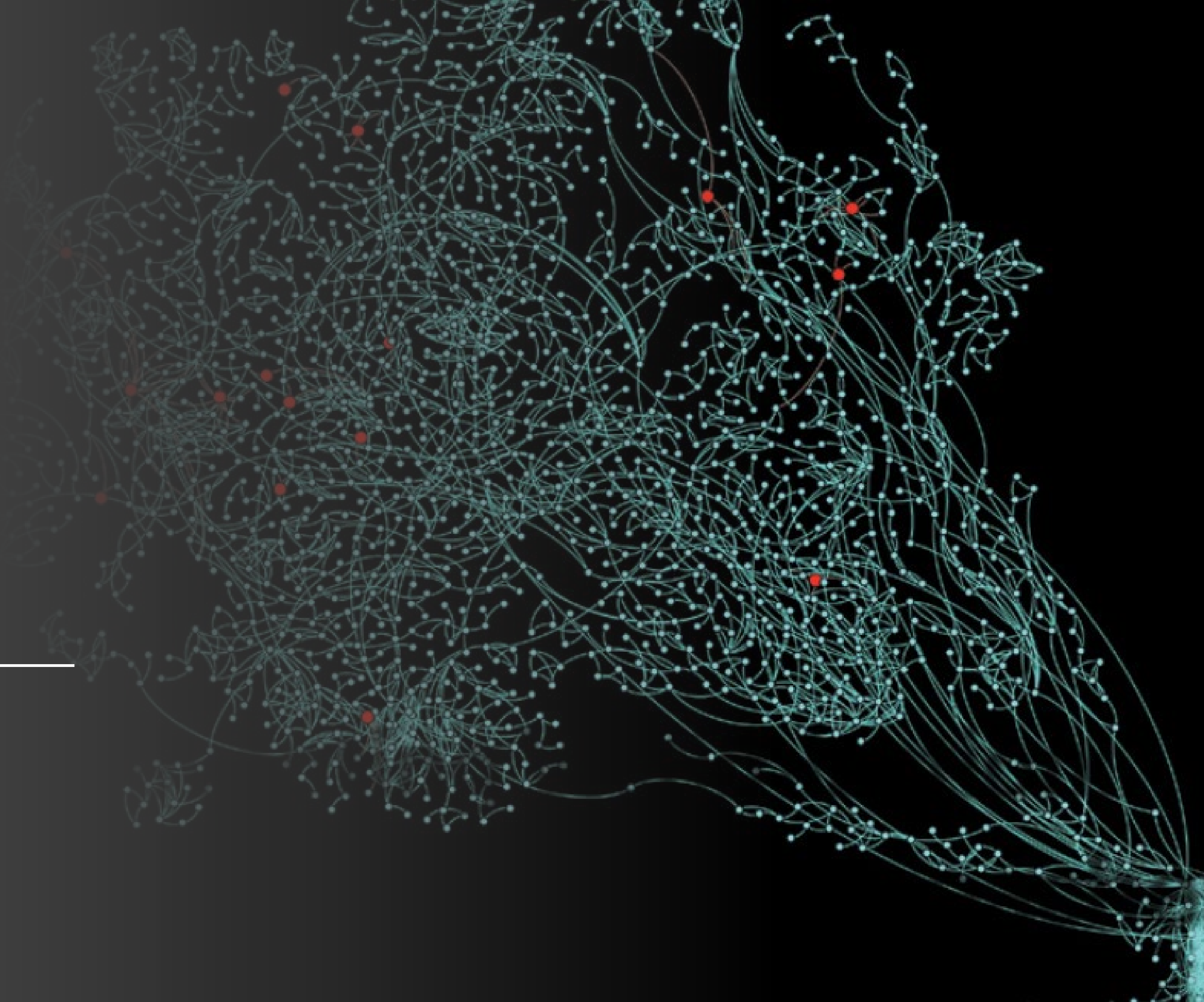
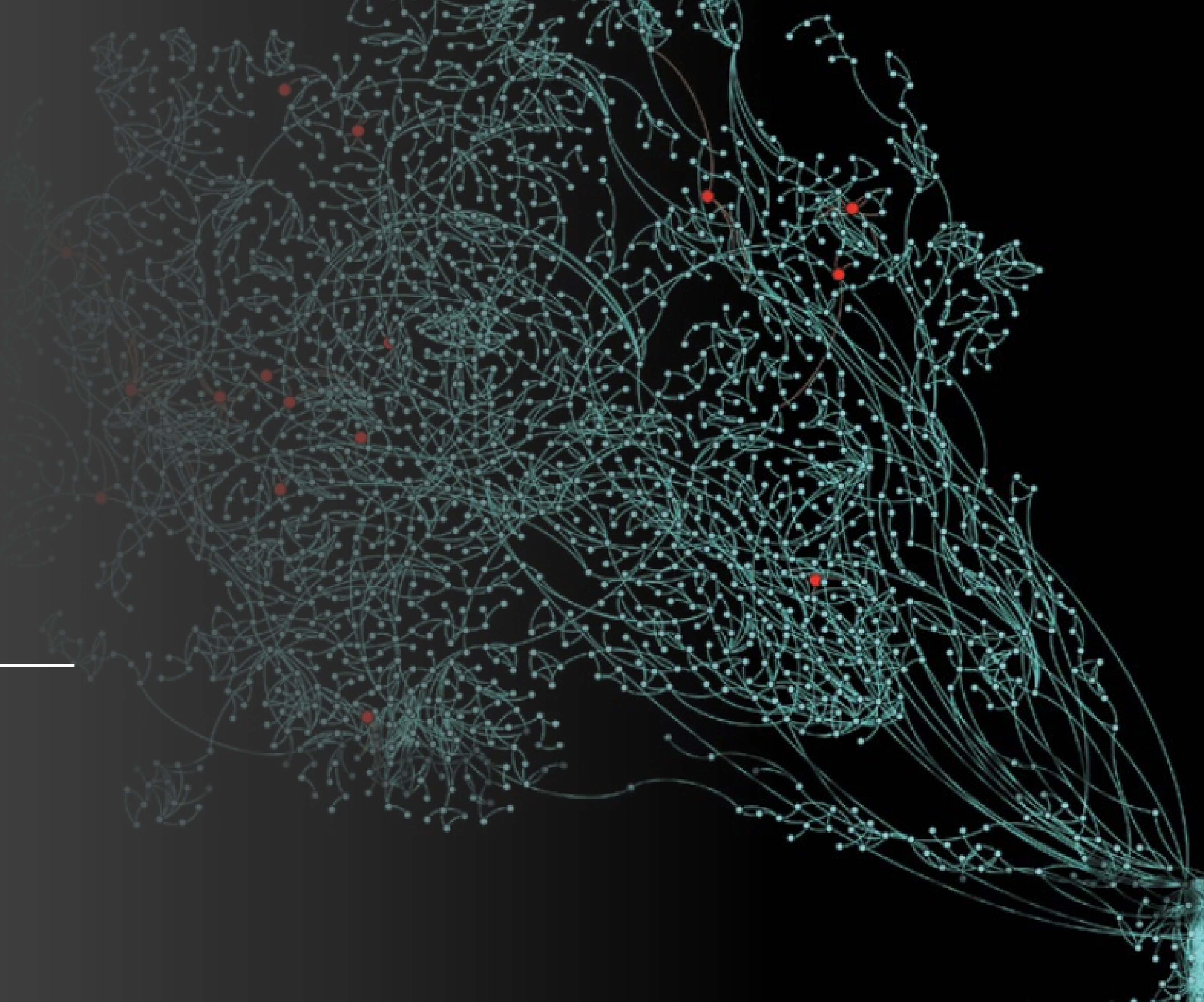
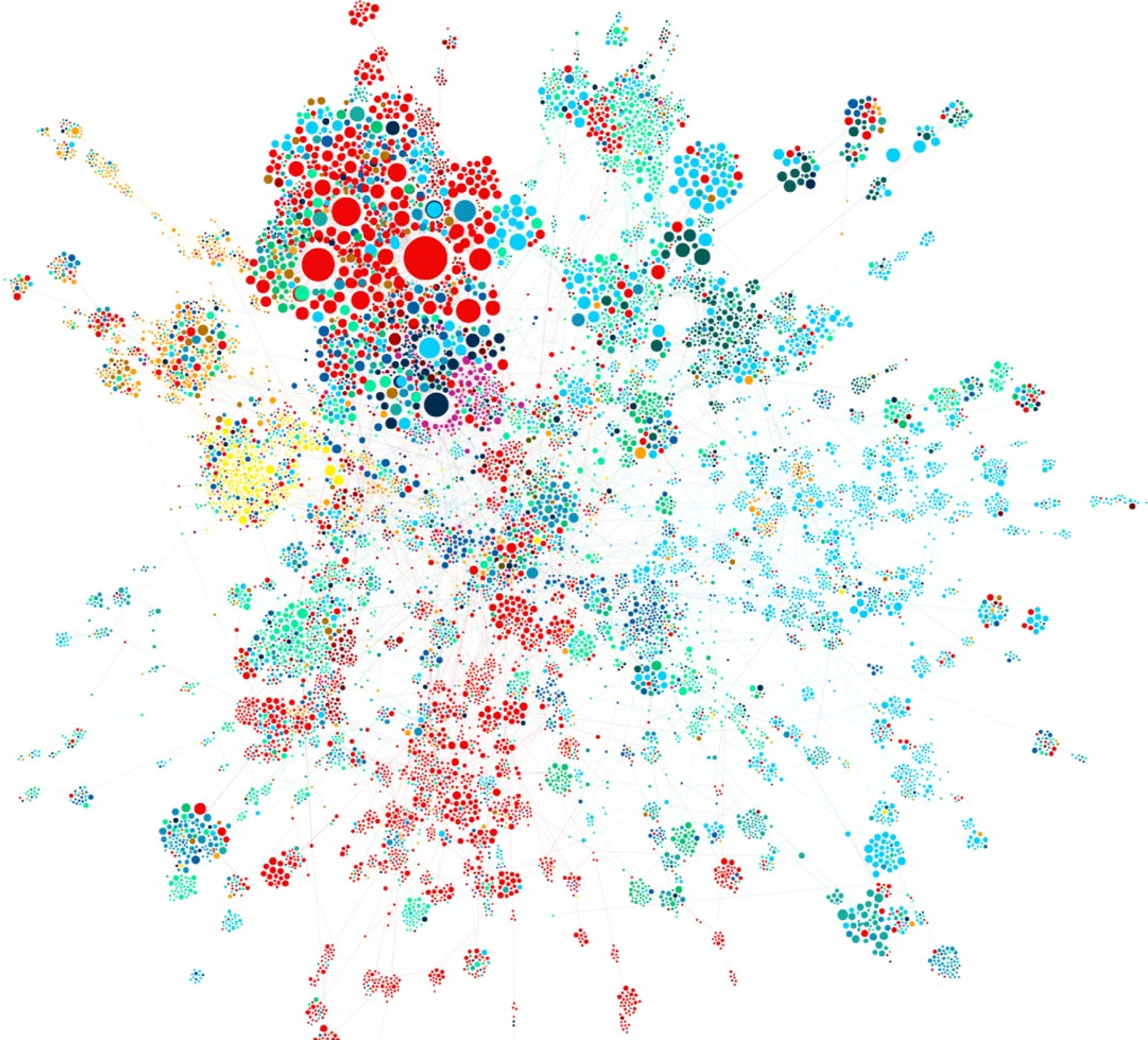
Network motifs



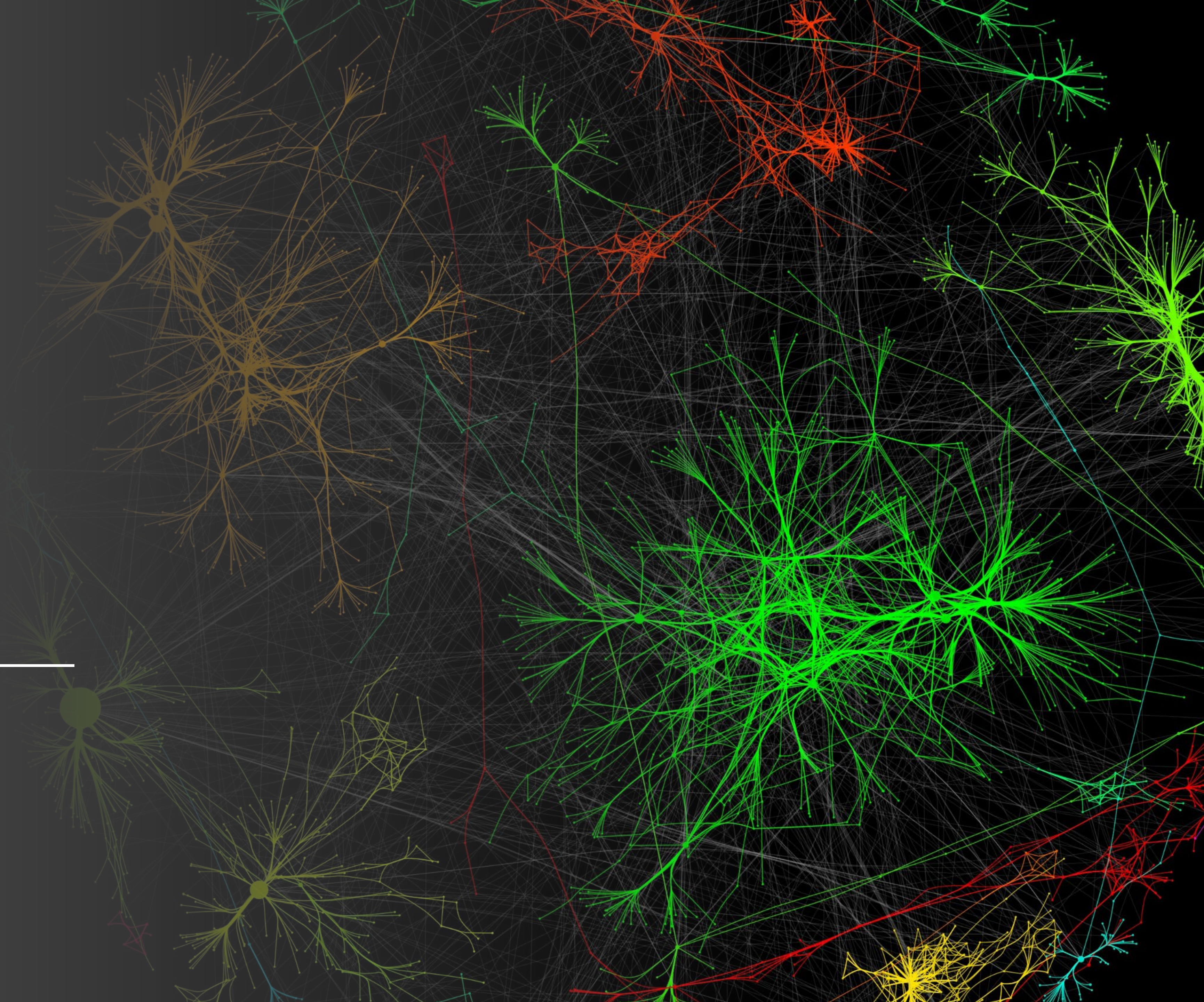
Social networks

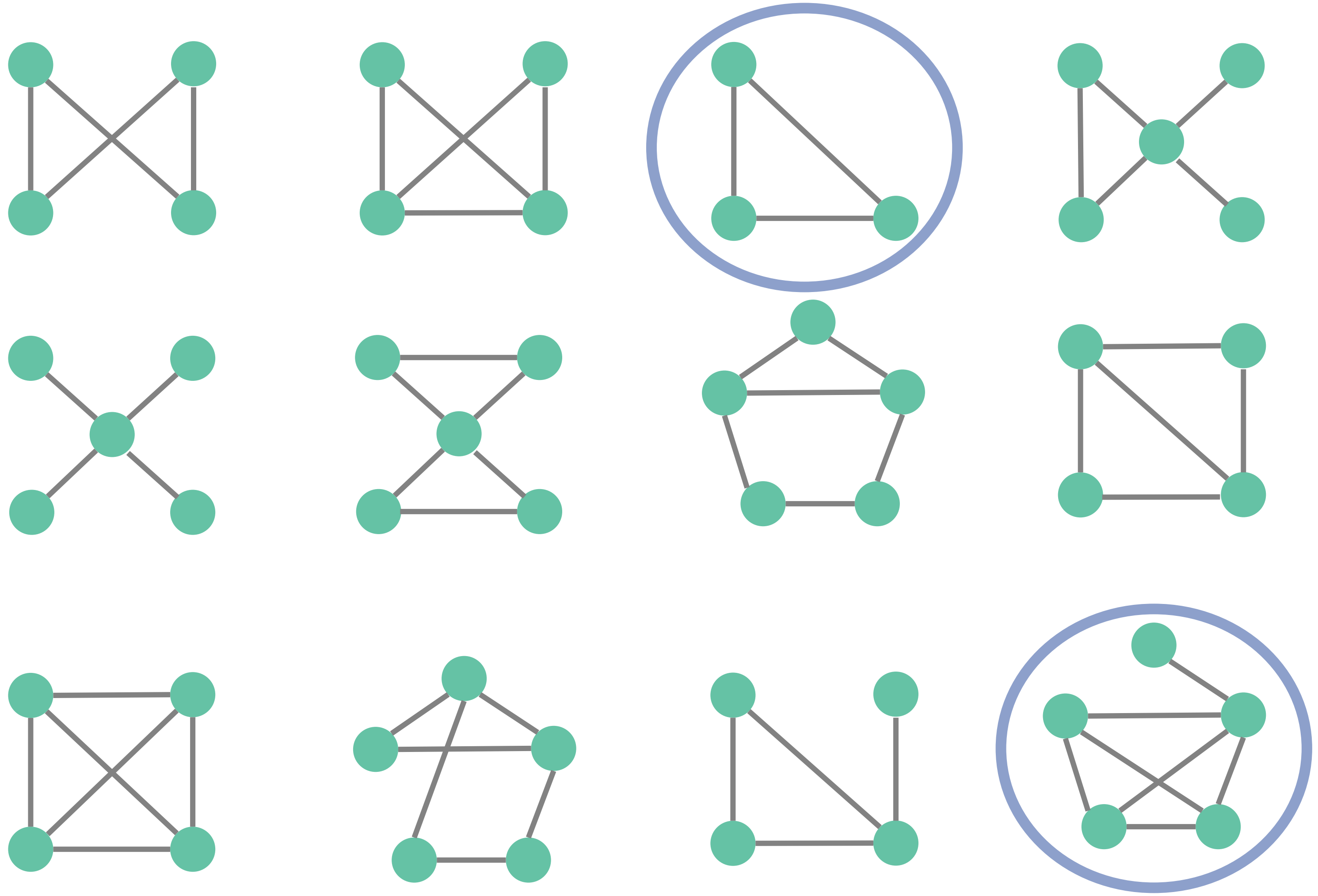


Collaboration networks

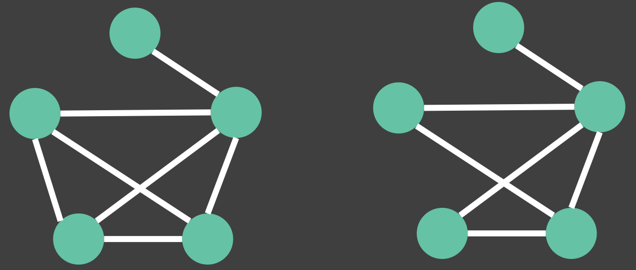


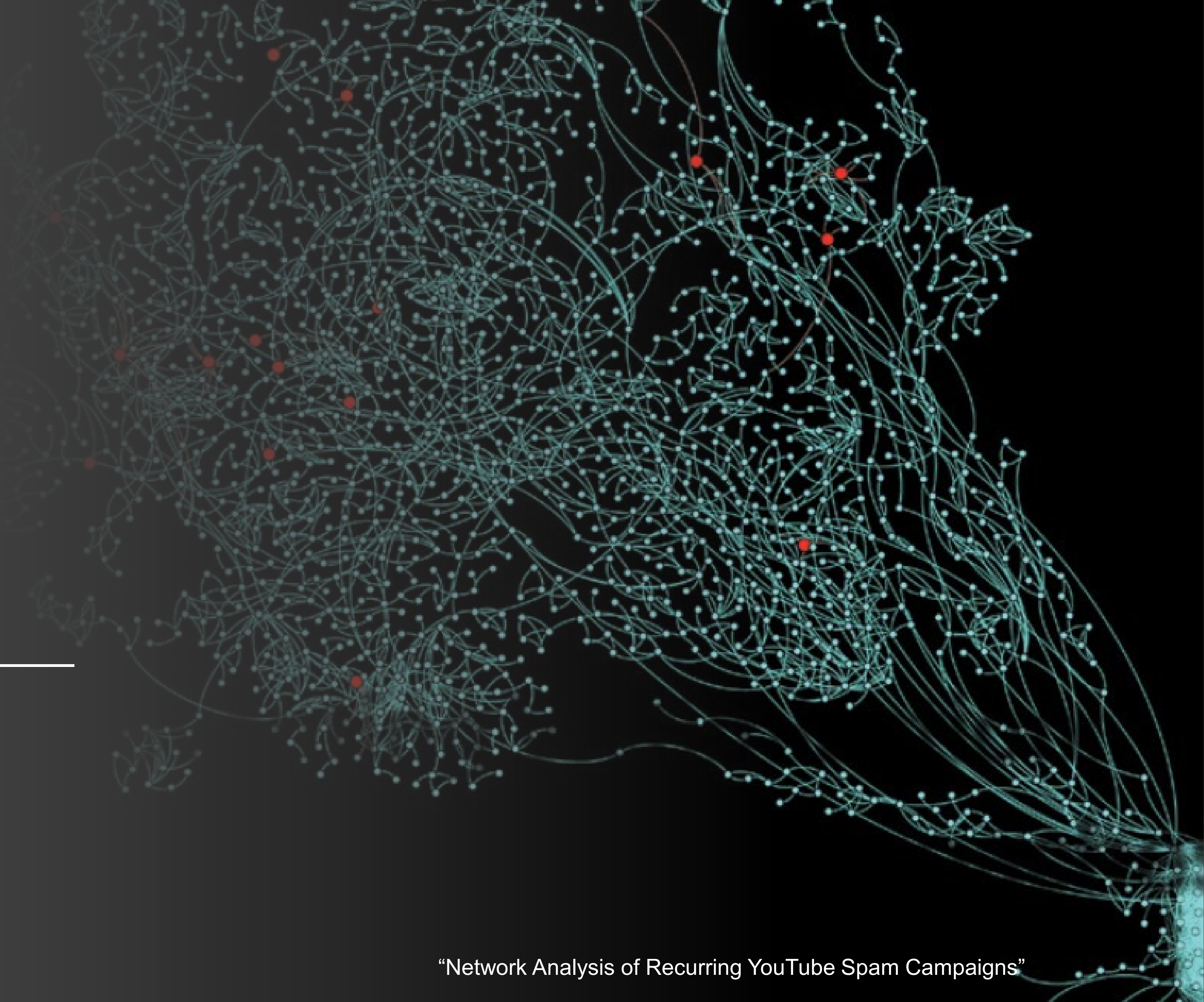
Protein networks





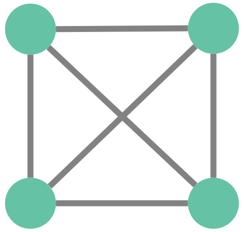
Trolls/spam

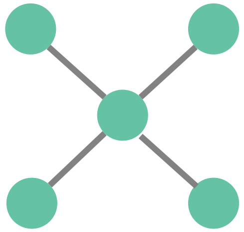


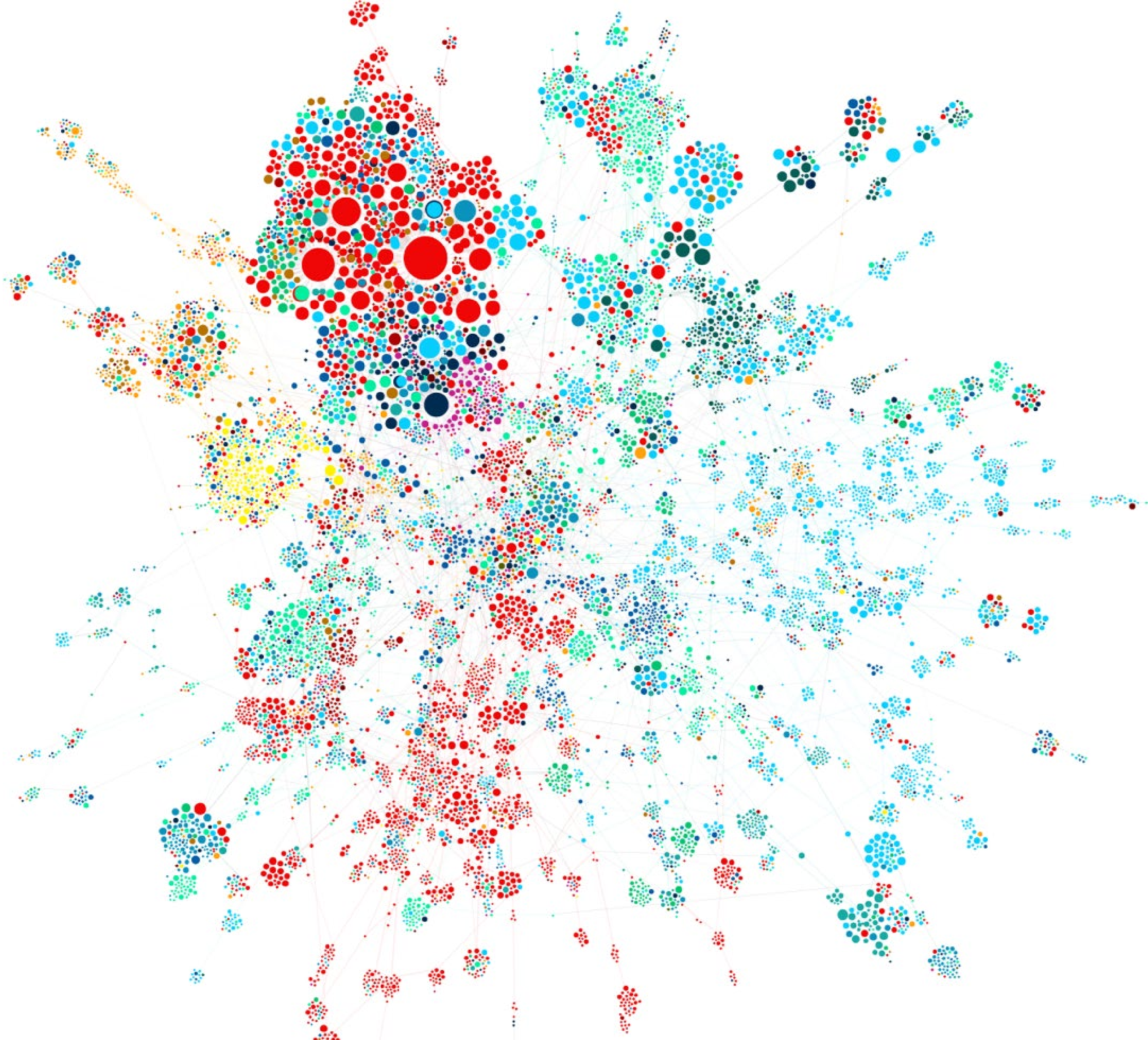


“Network Analysis of Recurring YouTube Spam Campaigns”

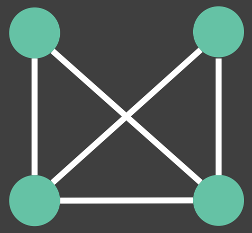
Predict success

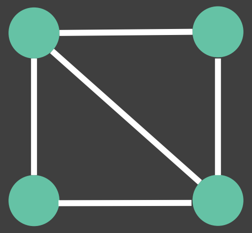




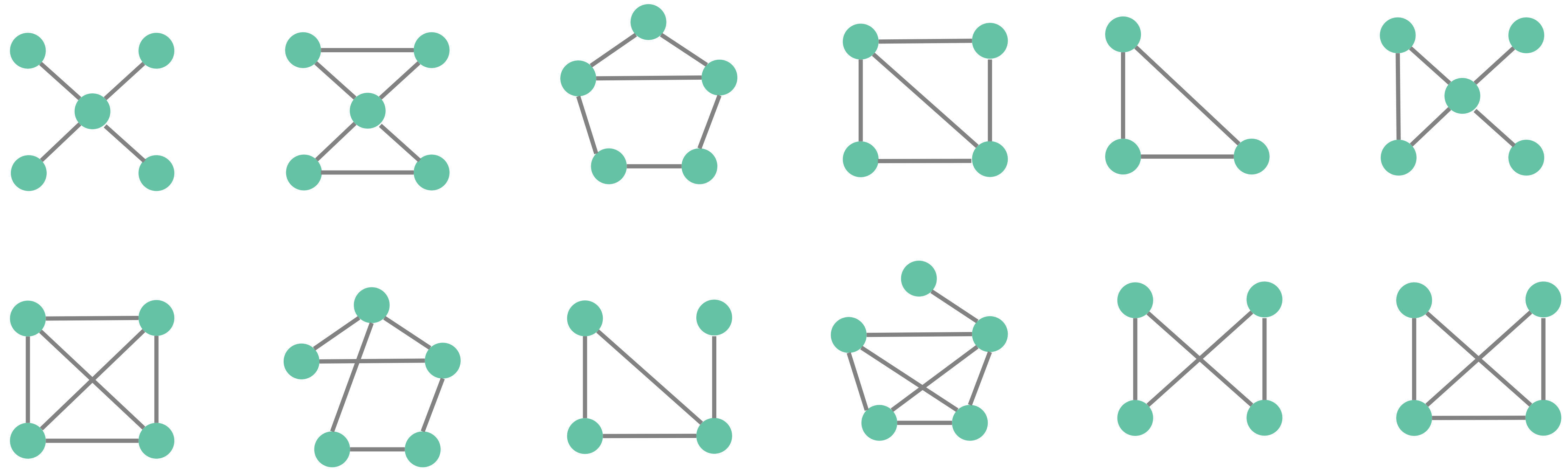


Genes with mutations





Motif = subgraph that appears more often than expected



Which motifs are significant?

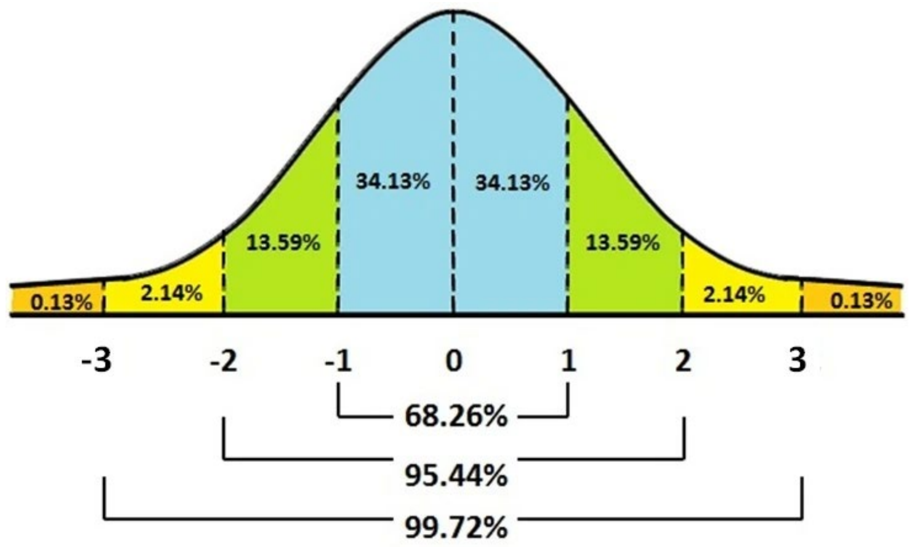
Network	Number of triangles	Number of claws	Number of 4 - cycles
David copperfield adjacent nouns	284	39.977	2.579
Catster social network	185.462.177	50.615.774.277	427.574.757.984
ArXiv collaborations	1.478.735	8.172.939.577	63.698.507

Which motifs are significant?

Network	*n*	Avg. degree	Max degree	Number of triangles	Number of claws	Number of 4 - cycles
David copperfield adjacent nouns	112	7	29	284	39.977	2.579
Catster social network	149.700	72	80.635	185.462.177	50.615.774.277	427.574.757.984
ArXiv collaborations	27.770	25	2.468	1.478.735	8.172.939.577	63.698.507

Z-score

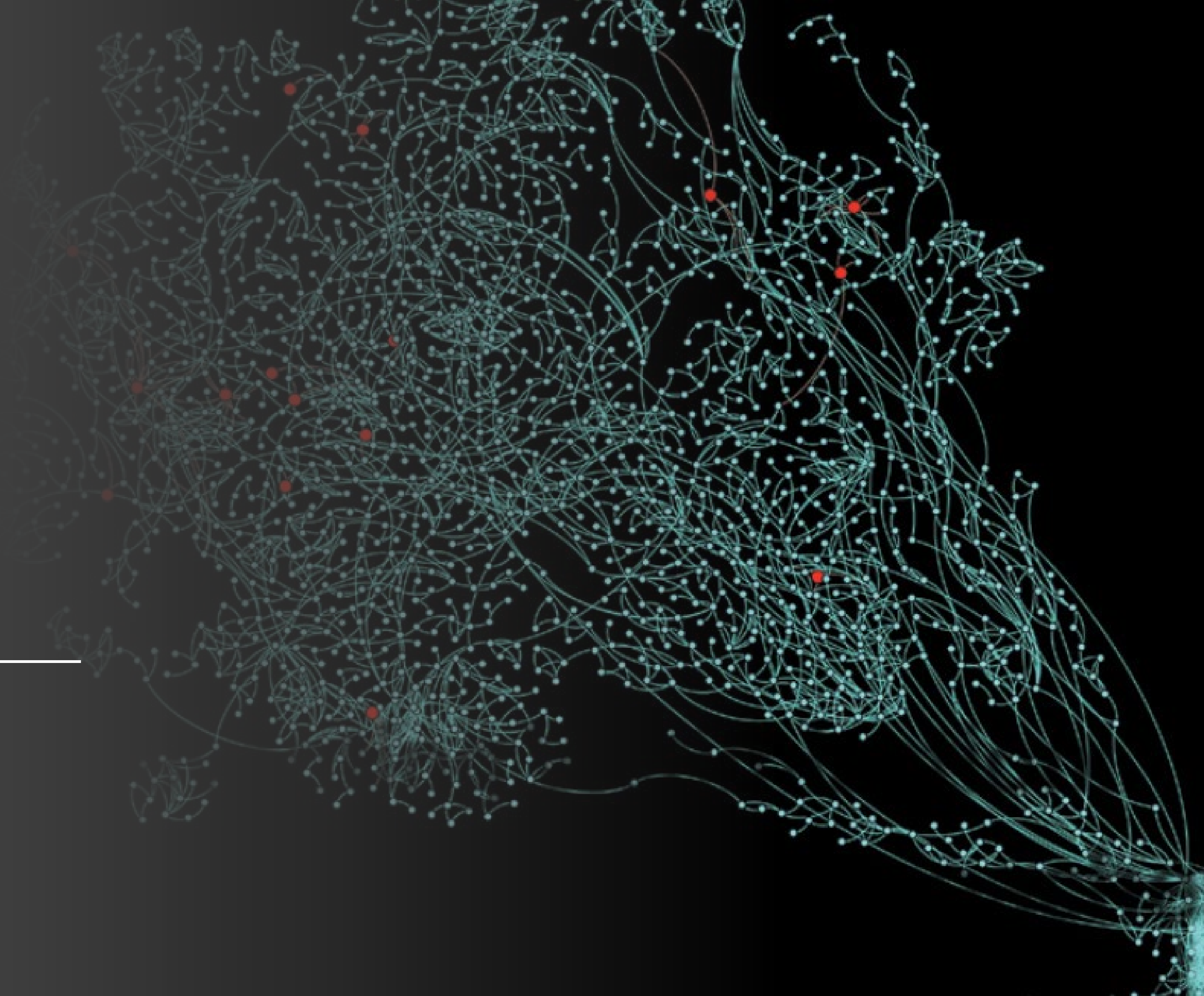
$$\frac{N_{H,data} - E[N_{H,null\ model}]}{\sqrt{Var}(N_{H,null\ model})}$$



How many standard deviations is $N_{H,data}$ away from the mean?

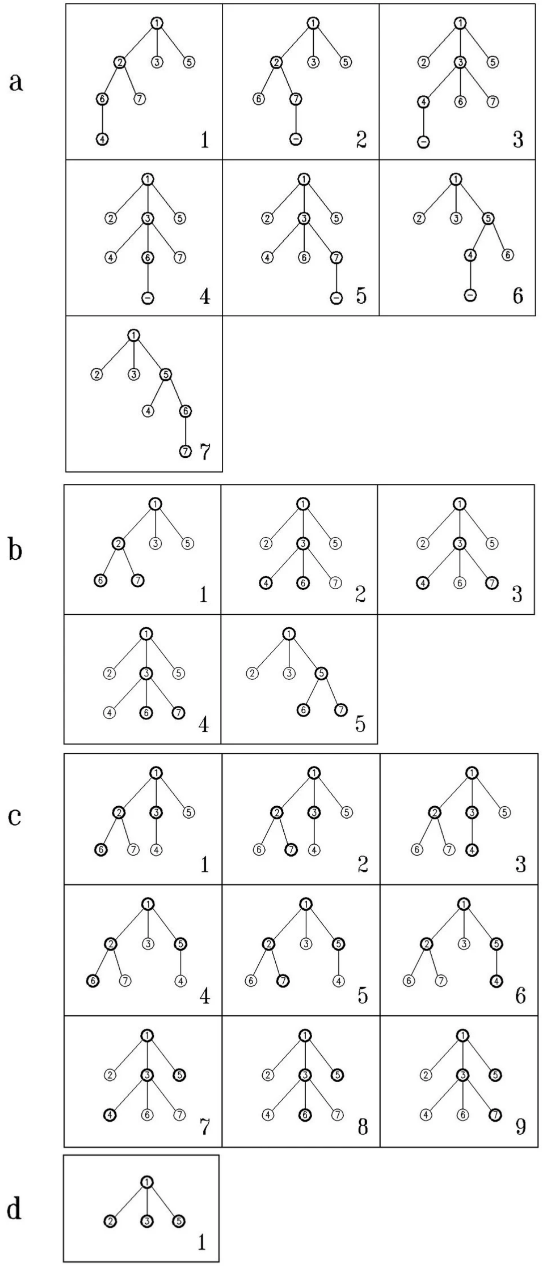
Significance is often measured assuming the normal distribution

Subgraph sampling



What if your network data is large?

- 'Naïve counting': $O(n^k)$ time for motifs of size k
- Better algorithms exist of $O(n^{1.5})$ for triangles
- Counting vs listing

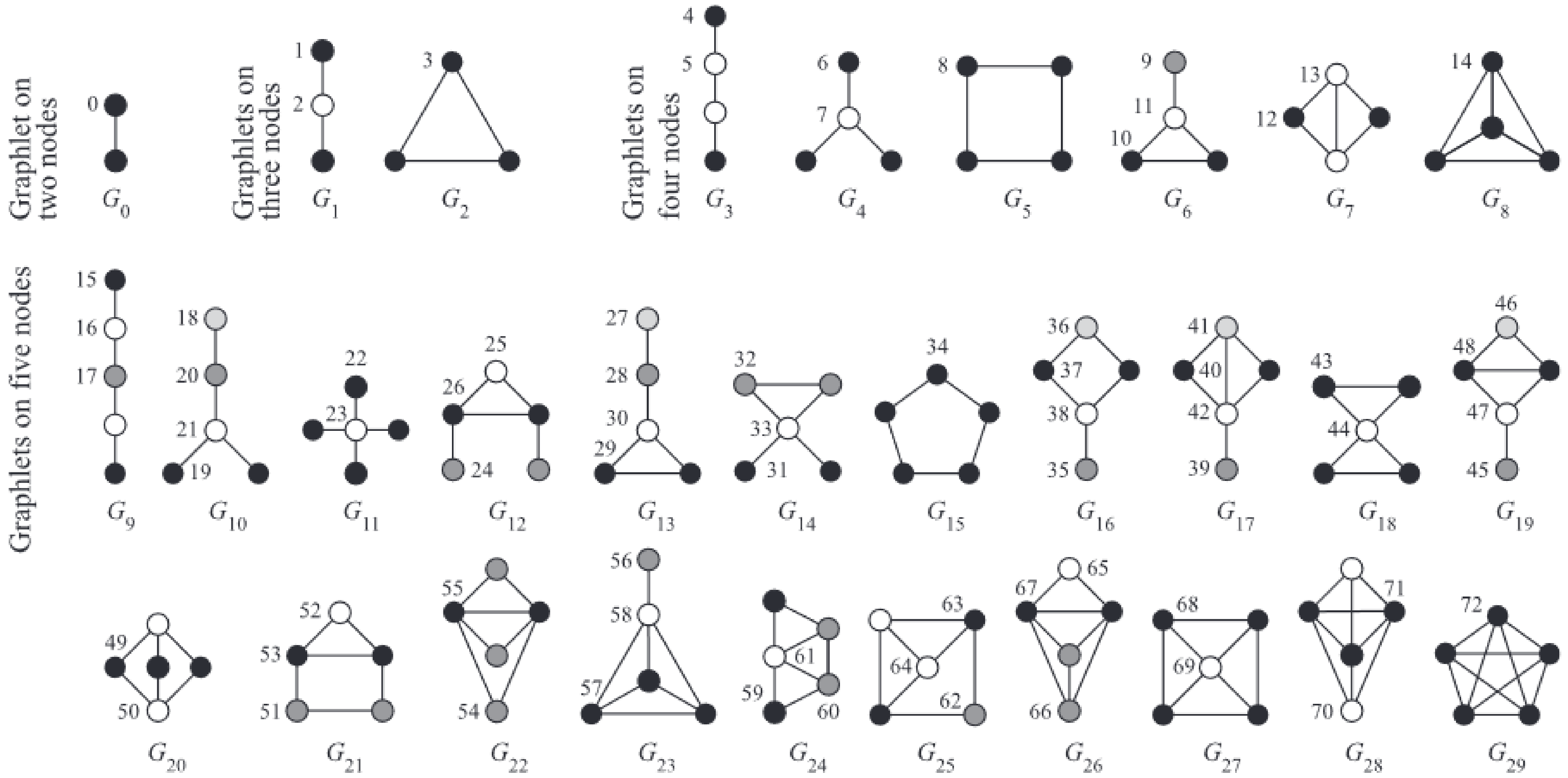


Combinatorial explosion

tech-as-skitter graph: 11M edges, but 2 trillion 5-cycles

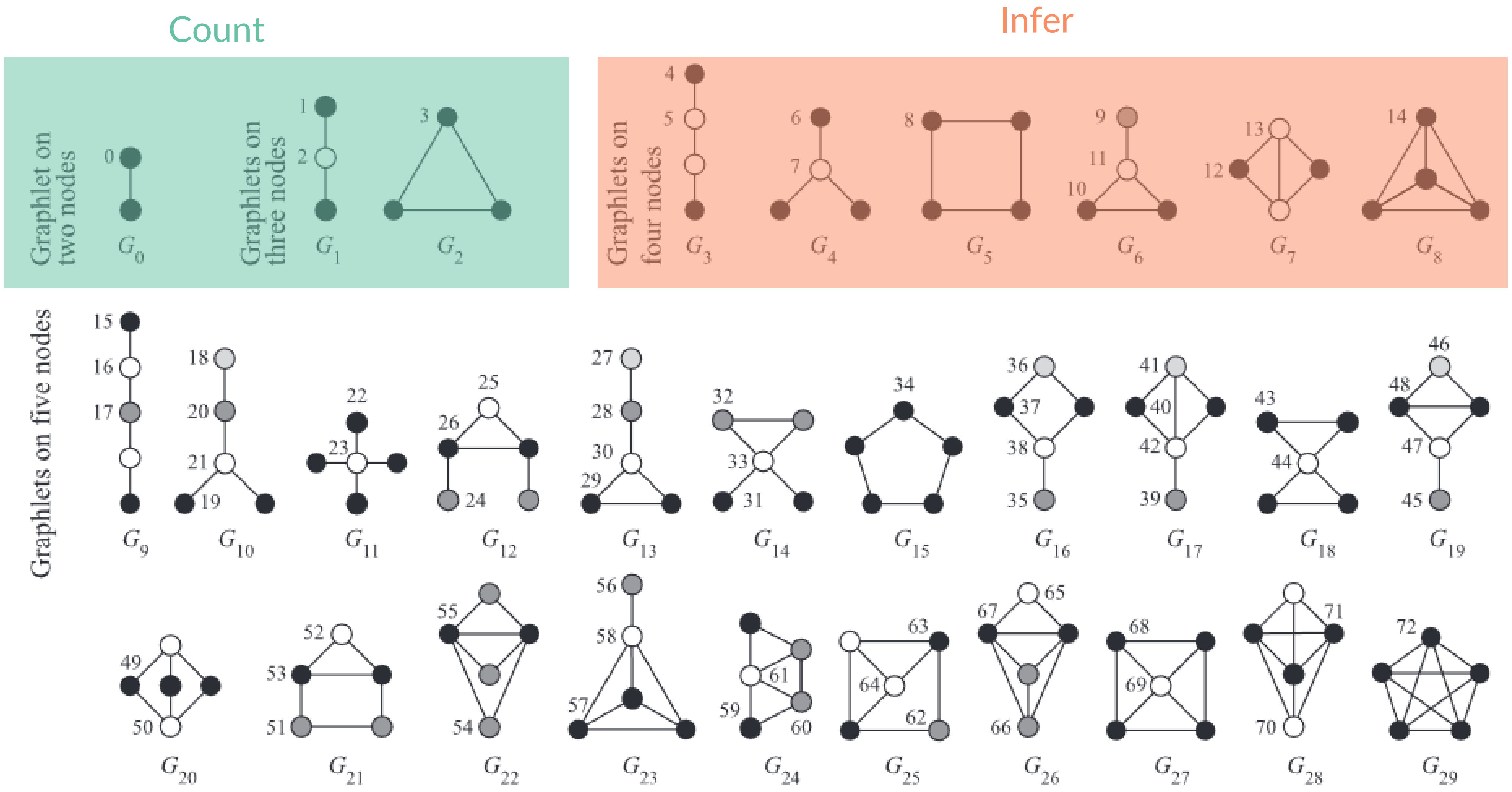
Listing is not possible

Orbit-based methods



A combinatorial approach to graphlet counting, 2014

Orbit-based methods

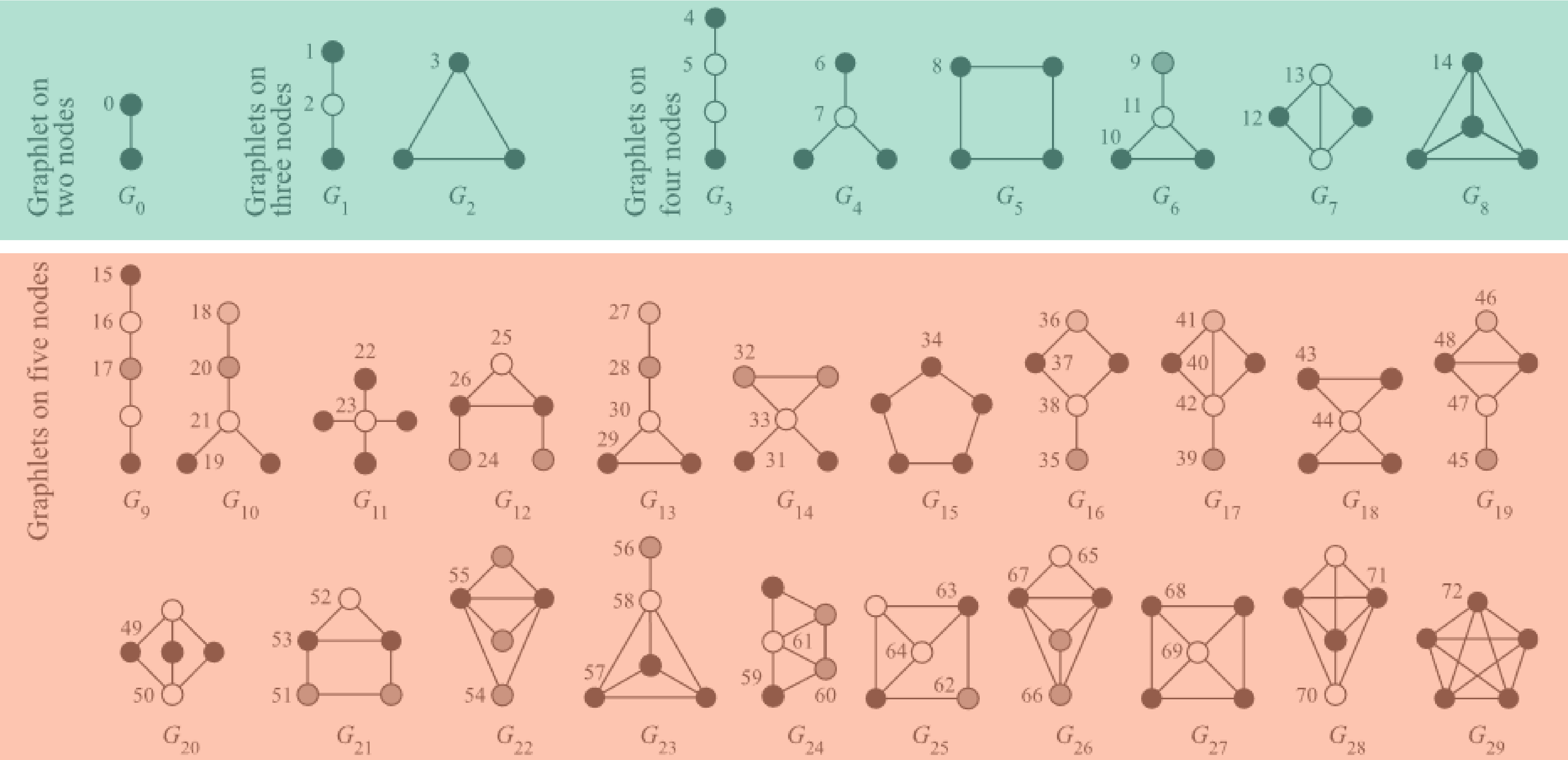


A combinatorial approach to graphlet counting, 2014

Orbit-based methods

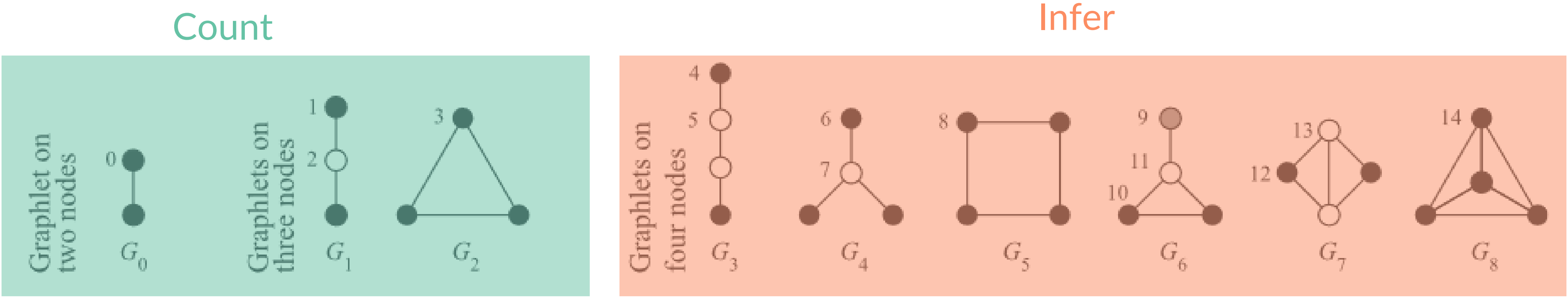
Count

Infer



A combinatorial approach to graphlet counting, 2014

Orbit-based methods



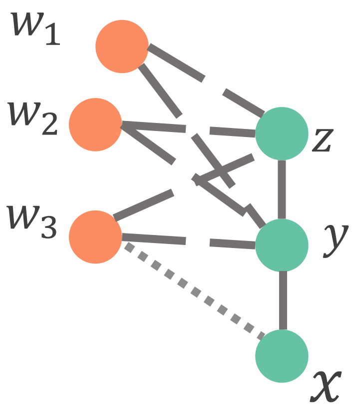
$o_i(x)$: Number of times node x has role i

Every 4-vertex subgraph can be created from a 3-vertex subgraph by adding a node

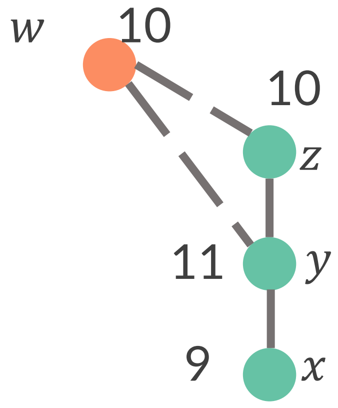
A combinatorial approach to graphlet counting, 2014

Relating orbits

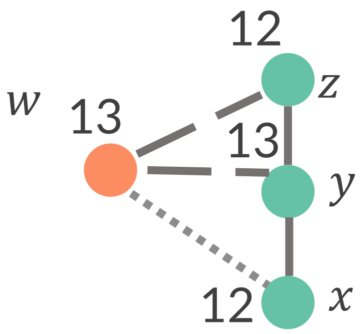
$$2o_9 + 2o_{12} = \sum_{y,z:G[x,y,z]\cong G_1} c(y,z)$$



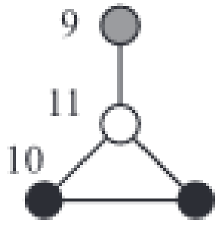
$c(y,z) = 3$



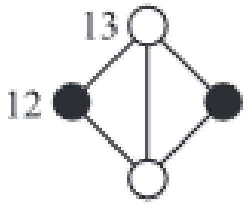
G_6



G_7



G_6

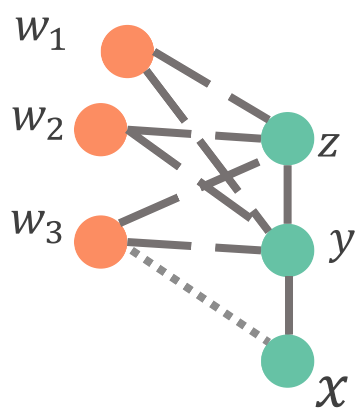


G_7

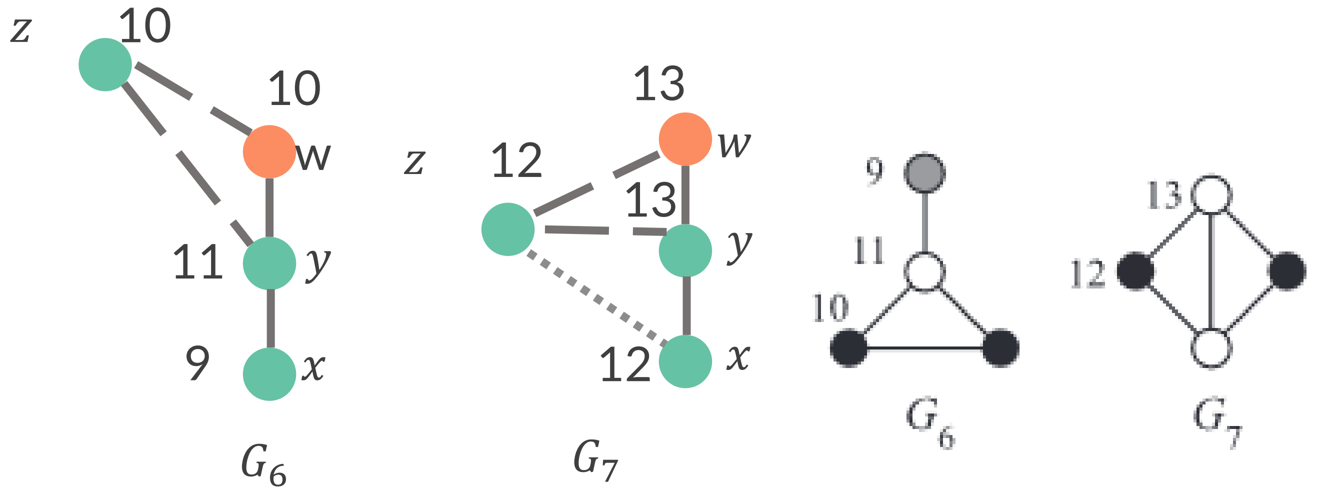
$c(y,z) =$ Number of triangles including edge y,z

Relating orbits

$$2o_9 + 2o_{12} = \sum_{y,z:G[x,y,z]\cong G_1} c(y,z)$$



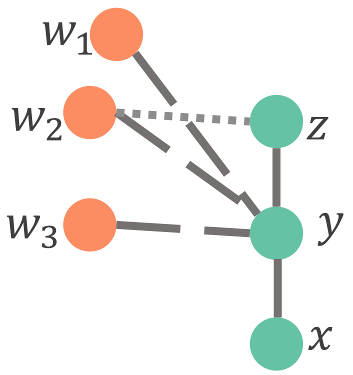
$c(y,z) = 3$

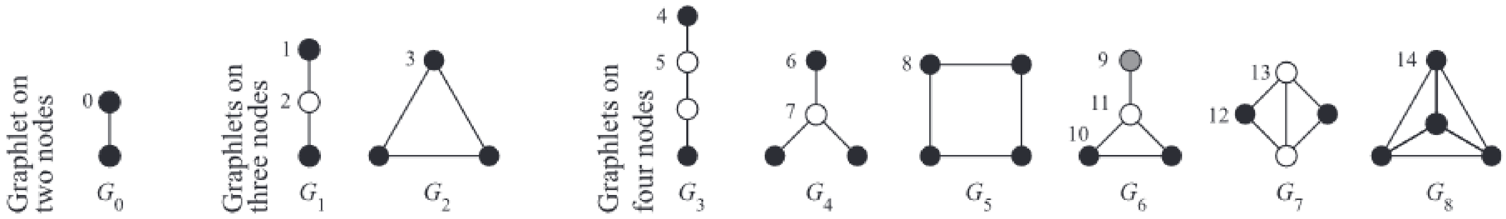


$c(y,z) =$ Number of triangles including y, z

Relate o_6 and o_9

$p(x, y)$: Number of paths (G_1) that start with nodes x, y





$\mathrm{p}(x, y) - 1 = 3$

Now try it yourself! (Exercise 1 + bonus)

With Jupyter Notebooks:

github.com/clarastegehuis/Complex_Networks_applications_school

Download folder and run Jupyter notebook

Without Jupyter Notebooks (with google account)

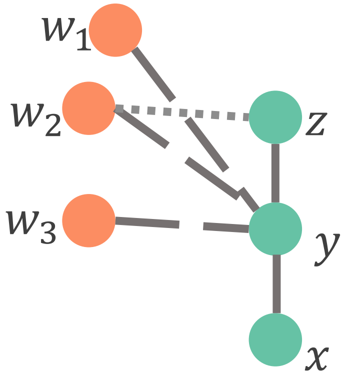
https://colab.research.google.com/github/clarastegehuis/Complex_Networks_applications_school

log in with Google account and run notebook

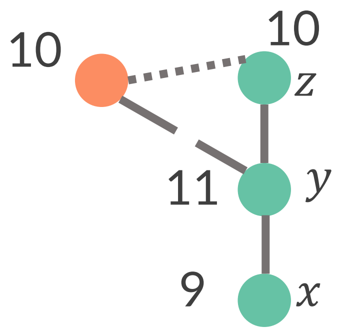
Relate o_6 and o_9

$p(x, y)$: Number of paths (G_1) that start with nodes x, y

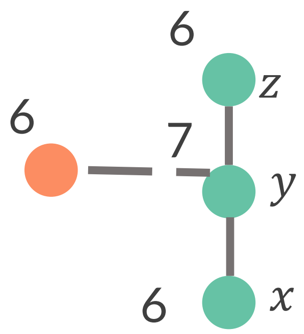
$$2o_6 + 2o_9 = \sum_{y,z:x,y,z=G_1} p(x,y) - 1$$



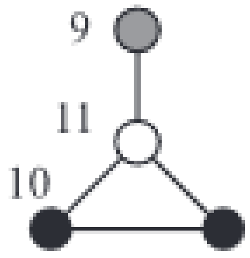
$p(x, y) - 1 = 3$



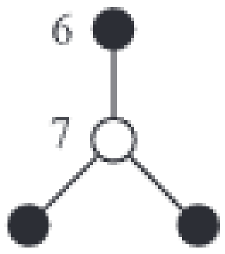
G_6



G_4



G_6

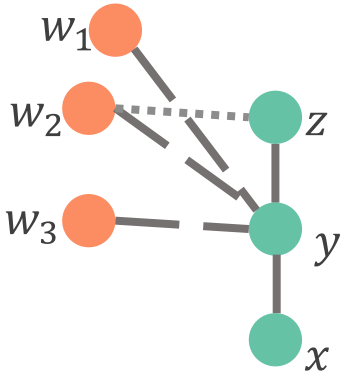


G_4

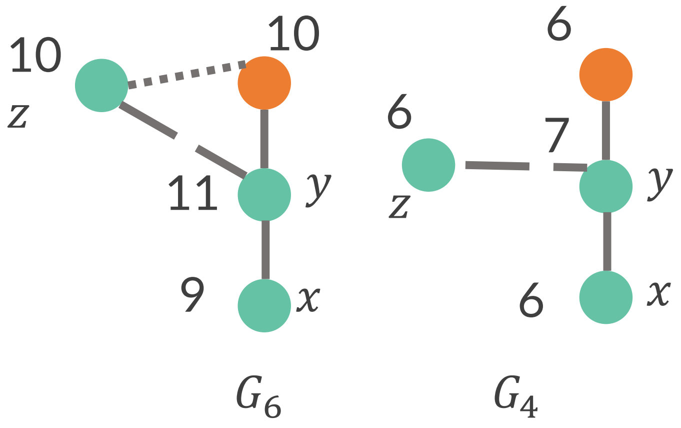
Relate o_6 and o_9

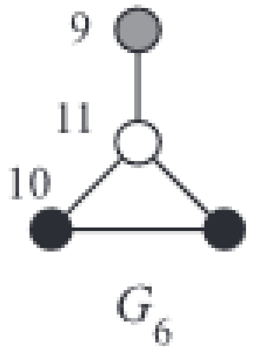
$p(x,y)$: Number of paths (G_1) that start with nodes x, y

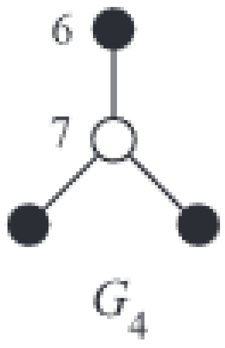
$$2o_6 + 2o_9 = \sum_{y,z:x,y,z=G_1} p(x,y) - 1$$



$\mathrm{p}(x,y) - 1 = 3$



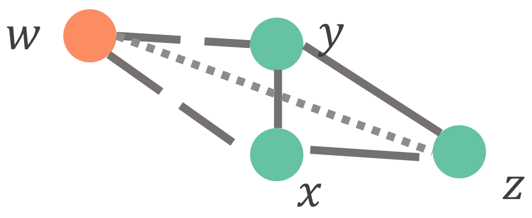


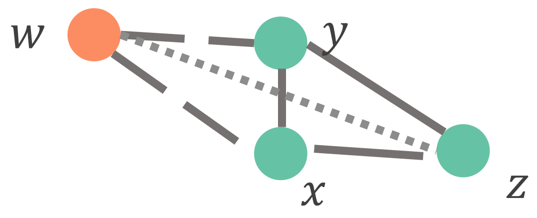


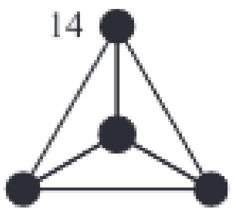
Relate o_{13} and o_{14}

$c(x, y)$: Number of common neighbors of x, y

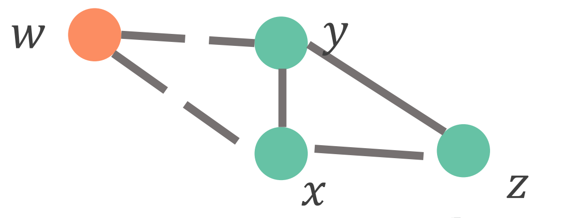
$$2o_{13} + 6o_{14} = \sum\nolimits_{y,z:x,y,z=G_2} c(x,y) - 1 + c(x,z) - 1$$

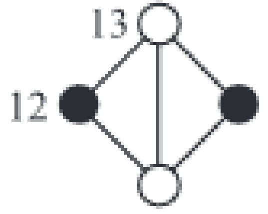






G_8



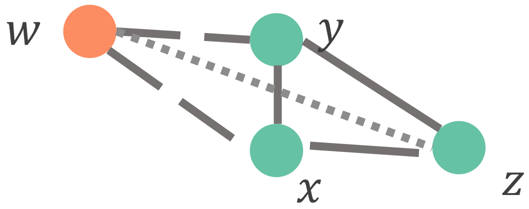


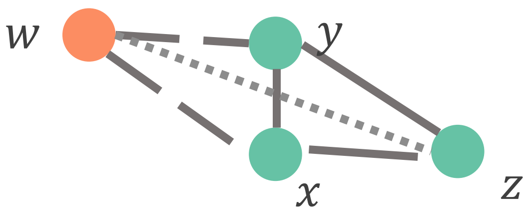
G_7

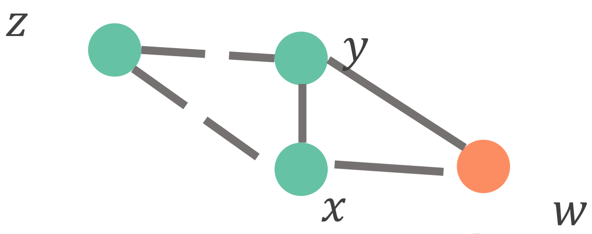
Relate o_{13} and o_{14}

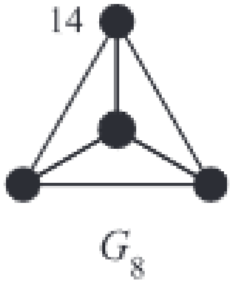
$c(x, y)$: Number of common neighbors of x, y

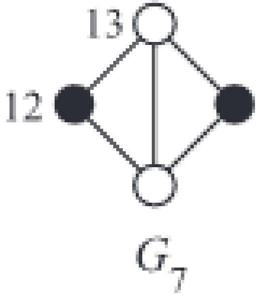
$$2o_{13} + 6o_{14} = \sum_{y,z:x,y,z=G_2} c(x,y) - 1 + c(x,z) - 1$$







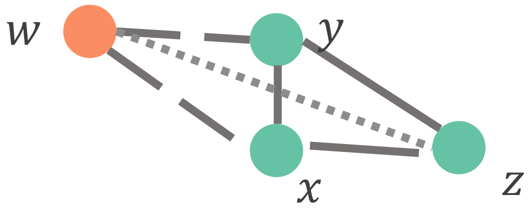


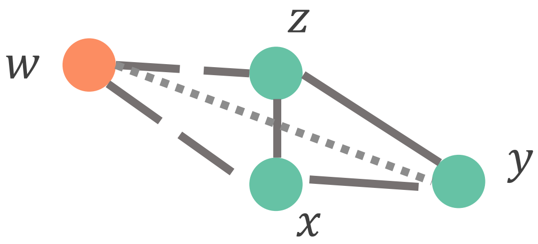


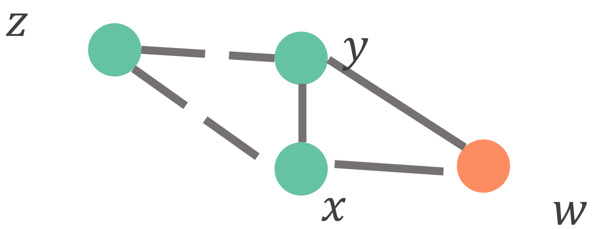
Relate o_{13} and o_{14}

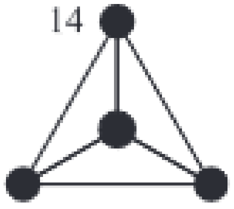
$c(x, y)$: Number of common neighbors of x, y

$$2o_{13} + 6o_{14} = \sum_{y,z:x,y,z=G_2} c(x,y) - 1 + c(x,z) - 1$$

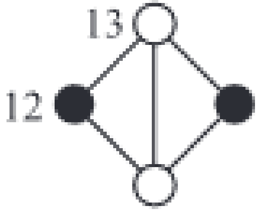








G_8

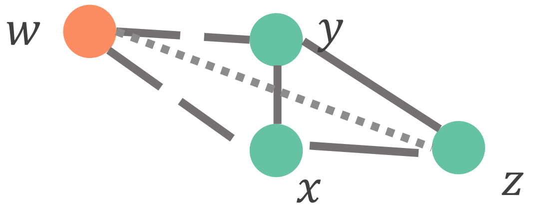


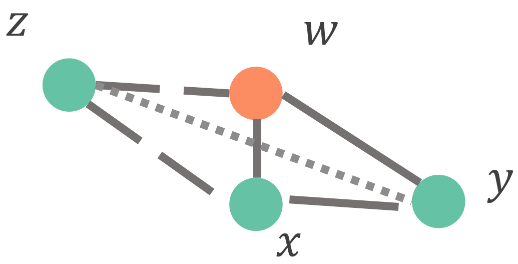
G_7

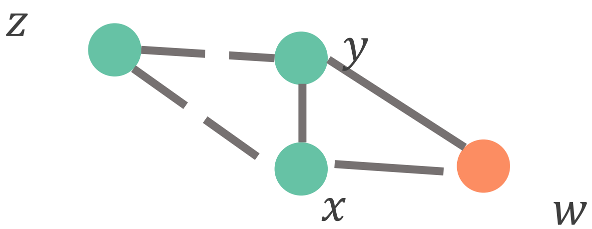
Relate o_{13} and o_{14}

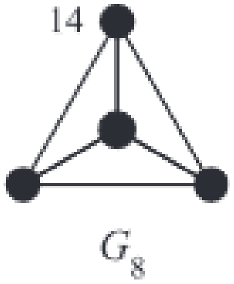
$c(x, y)$: Number of common neighbors of x, y

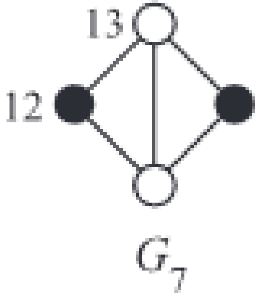
$$2o_{13} + 6o_{14} = \sum_{y,z:x,y,z=G_2} c(x,y) - 1 + c(x,z) - 1$$











Obtain per-node orbit counts

To get size-4 orbits, compute:

- $p(x, y)$ and $c(x, y)$
- one orbit count.

Worst-case time complexity:
$O(nd + nd^3)$

10 equations

Equations involving 3-node subgraphs

11 orbits (unknown)

$$o_{12} + 3o_{14} = \sum_{y,z:\, y<z,\, G[\{x,y,z\}] \cong G_2} c(y,z) - 1$$

$$2o_{13} + 6o_{14} = \sum_{y,z:\, y<z,\, G[\{x,y,z\}] \cong G_2} (c(x,y) - 1) + (c(x,z) - 1)$$

$$o_{10} + 2o_{13} = \sum_{y,z:\, y<z,\, G[\{x,y,z\}] \cong G_2} p(y,z) + p(z,y)$$

$$2o_{11} + 2o_{13} = \sum_{y,z:\, y<z,\, G[\{x,y,z\}] \cong G_2} p(y,x) + p(z,x)$$

$$6o_7 + 2o_{11} = \sum_{y,z:\, y<z,\, y,z \in N(x),\, G[\{x,y,z\}] \cong G_1} (p(y,x) - 1) + (p(z,x) - 1)$$

$$o_5 + 2o_8 = \sum_{y,z:\, y<z,\, y,z \in N(x),\, G[\{x,y,z\}] \cong G_1} p(x,y) + p(x,z)$$

$$2o_6 + 2o_9 = \sum_{y,z:\, x,z \in N(y),\, G[\{x,y,z\}] \cong G_1} p(x,y) - 1$$

$$2o_9 + 2o_{12} = \sum_{y,z:\, x,z \in N(y),\, G[\{x,y,z\}] \cong G_1} c(y,z)$$

$$o_4 + 2o_8 = \sum_{y,z:\, x,z \in N(y),\, G[\{x,y,z\}] \cong G_1} p(y,z)$$

$$2o_8 + 2o_{12} = \sum_{y,z:\, x,z \in N(y),\, G[\{x,y,z\}] \cong G_1} c(x,z) - 1$$

State of the art

- Counting 4-vertex subgraphs:

  For 117M edge social graph 22m on laptop (ESCAPE)
- Counting 5-vertex subgraphs:

  For graphs with 10M edges, less than 30 minutes

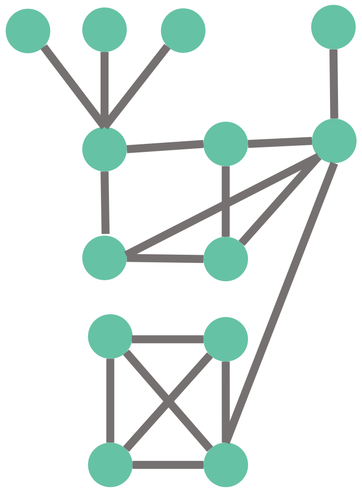
  For 117M edge social graph, 30 hours

Algorithm converts graph to directed, and uses fewer subgraphs

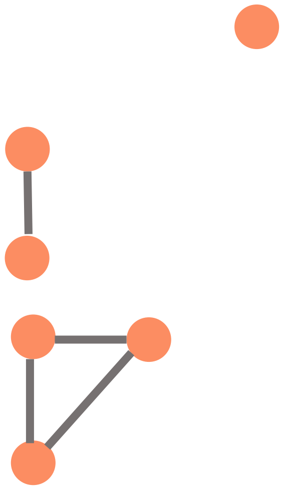
'ESCAPE: Efficiently Counting All 5-Vertex Subgraphs'

What if your network data is large?

- Approximate counting: subsample your network data
- Simplest method: keep every node with probability p
- Then count subgraphs



Original graph



Subsampled graph

What is the probability that a subgraph remains in the sampled data?

Try it yourself!

Now try it yourself! (Part 2)

With Jupyter Notebooks:

github.com/clarastegehuis/Complex_Networks_applications_school

Download folder and run Jupyter notebook

Without Jupyter Notebooks (with google account)

https://colab.research.google.com/github/clarastegehuis/Complex_Networks_applications_school

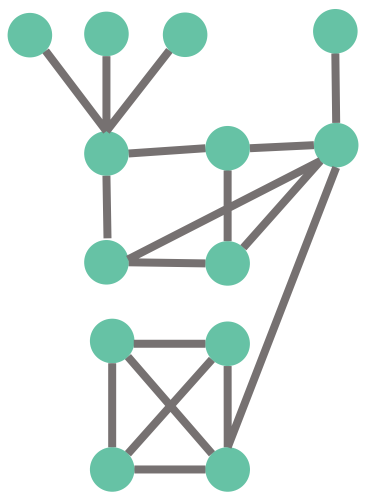
log in with Google account and run notebook

Any triangle remains a triangle in subsample with probability p^3

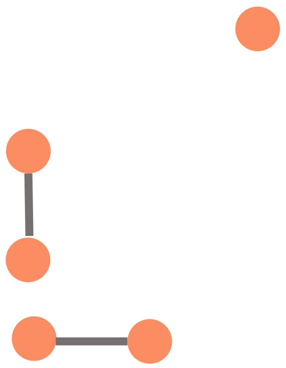
Thus, on average,

$$N_{\Delta} p^3 = N_{\Delta, subsample}$$

Disadvantage: many isolated nodes

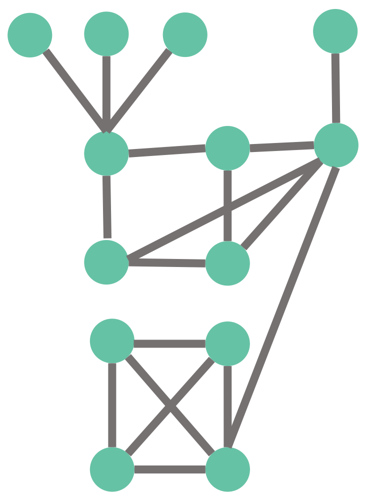


Original graph

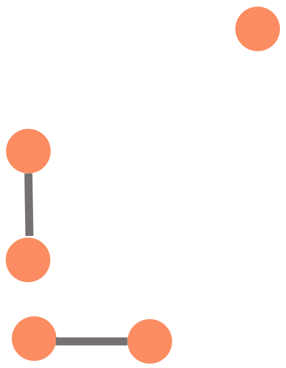


Random sampling

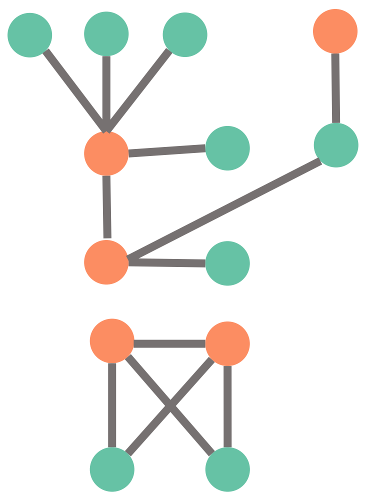
More advanced sampling methods



Original graph

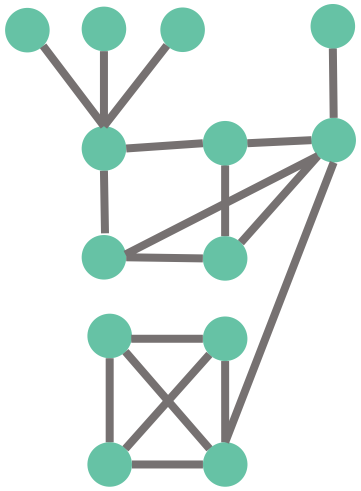


Random sampling

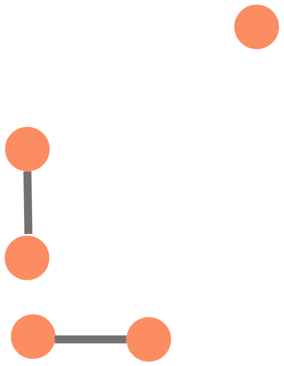


Neighborhood sampling

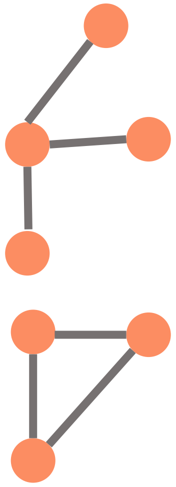
More advanced sampling methods



Original graph

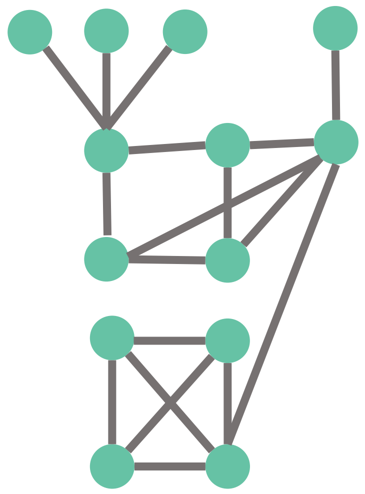


Random sampling

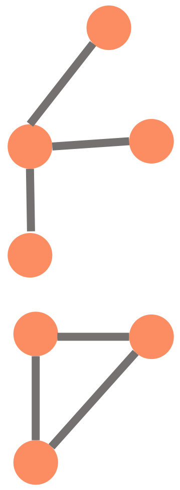


Random walk-based sampling

Edge sampling (ESA)



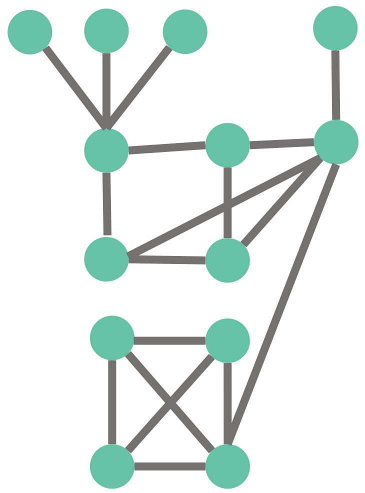
Original graph



Random neighborhood sampling

'Efficient sampling algorithm for estimating subgraph concentrations and detecting network motifs', Kashtan et al, 2004

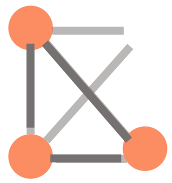
Edge sampling (ESA)



Original graph

What is the probability of sampling this triangle in this order?

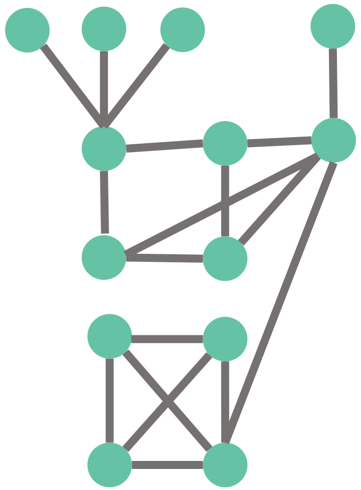
$$\frac{1}{18} * \frac{1}{4}$$



Random neighborhood sampling

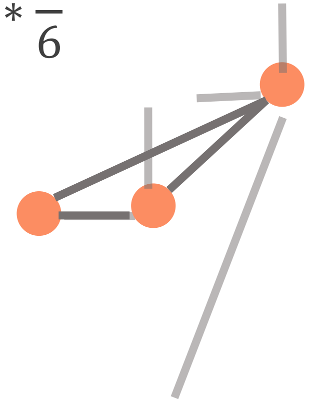
Edge sampling (ESA)

Is this probability the same for all triangles?



Original graph

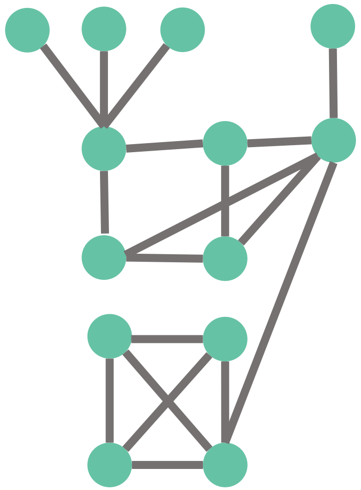
$$\frac{1}{18} * \frac{1}{6}$$



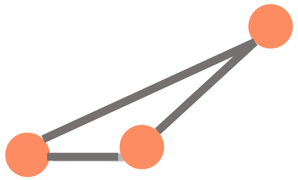
Random neighborhood sampling

Edge sampling (ESA)

Total probability to observe this triangle is averaged over all orderings



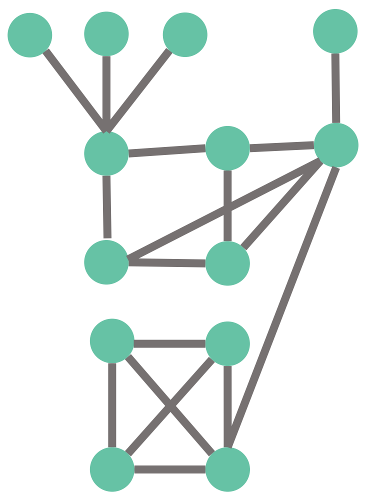
Original graph



Random neighborhood sampling

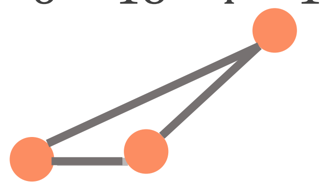
Edge sampling (ESA)

Total probability to observe this triangle is averaged over all orderings



Original graph

$$\frac{1}{18} * \frac{2}{6} + \frac{1}{18} * \frac{2}{4} + \frac{1}{18} * \frac{2}{5}$$



Random neighborhood sampling

Edge sampling (ESA)

Input: A graph $G = (V, E)$ and an integer $2 \leq k \leq |V|$.
Output: Vertices of a randomly chosen size-k subgraph in G.

$\{u, v\} \leftarrow$ random edge from E
$V' \leftarrow \{u, v\}$
while $|V'| \neq k$ **do**
    $\{u, v\} \leftarrow$ random edge from $V' \times N(V')$
    $V' \leftarrow V' \cup \{u\} \cup \{v\}$
return V'

Generate list L of sampled size-k subgraphs

$$\text{Estimated density of } H = \frac{\sum_{G \in L \mid G = H} P(G \text{ is sampled by } ESA)^{-1}}{\sum_{G \in L} P(G \text{ is sampled by } ESA)^{-1}}$$

Now try it yourself! (Part 3)

With Jupyter Notebooks:

github.com/clarastegehuis/Complex_Networks_applications_school

Download folder and run Jupyter notebook

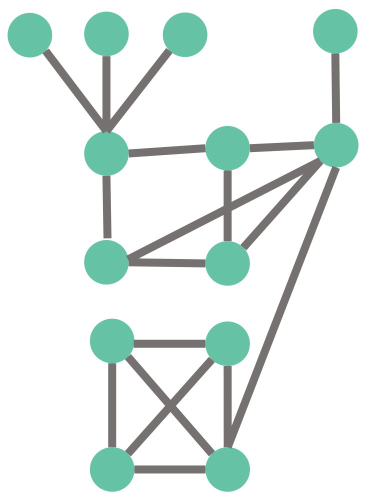
Without Jupyter Notebooks (with google account)

https://colab.research.google.com/github/clarastegehuis/Complex_Networks_applications_school

log in with Google account and run notebook

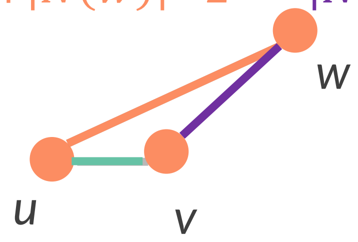
Edge sampling (ESA)

Total probability to observe triangle averaged over all orderings



Original graph

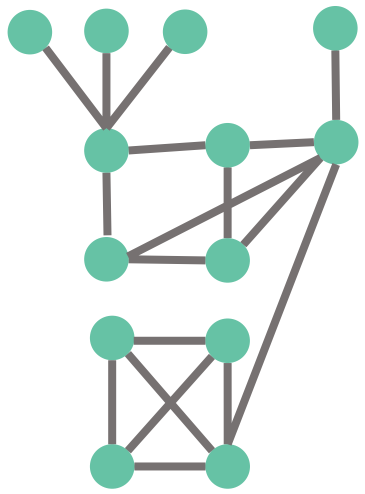
$$\frac{1}{|E|}\left(\frac{2}{|N(u)|+|N(v)|-2}+\frac{2}{|N(u)|+|N(w)|-2}+\frac{2}{|N(v)|+|N(w)|-2}\right)$$



Random neighborhood sampling

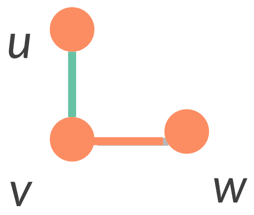
Edge sampling (ESA)

Total probability to observe wedge averaged over all orderings



Original graph

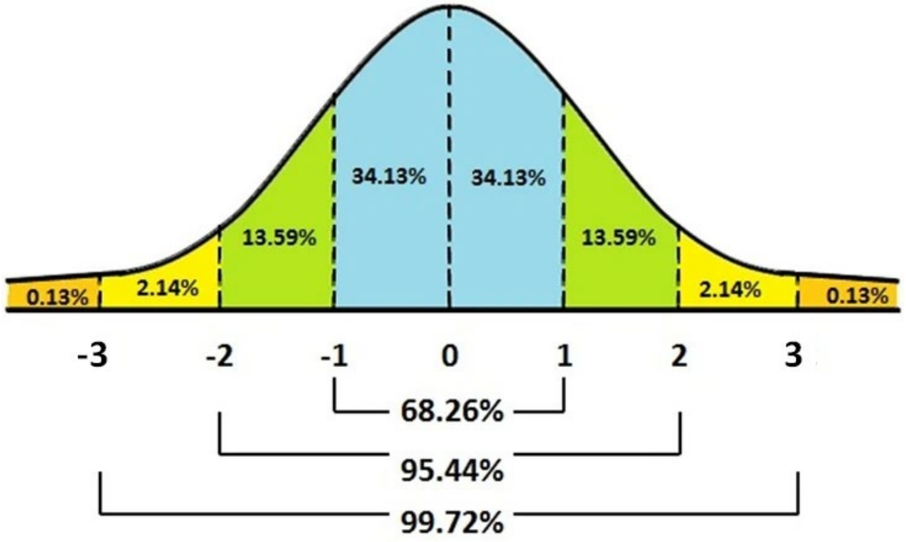
$$\frac{1}{|E|}\left(\frac{1}{|N(u)|+|N(v)|-2}+\frac{1}{|N(v)|+|N(w)|-2}\right)$$



Random neighborhood sampling

Z-score

$$\frac{N_{H,data} - E[N_{H,null\ model}]}{\sqrt{Var}(N_{H,null\ model})}$$

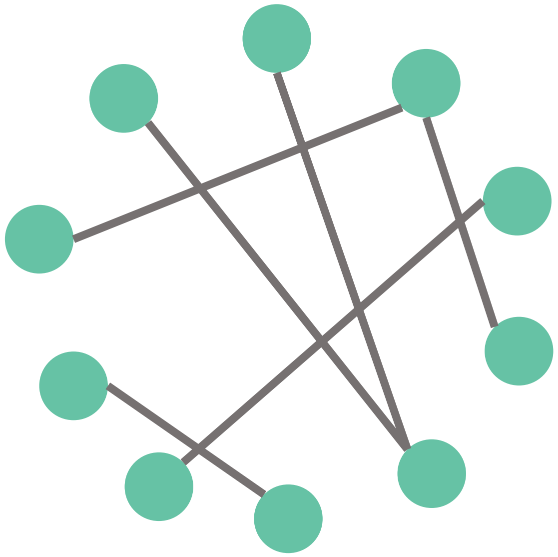


How many standard deviations is $N_{H,data}$ away from the mean?

Significance is often measured assuming the normal distribution

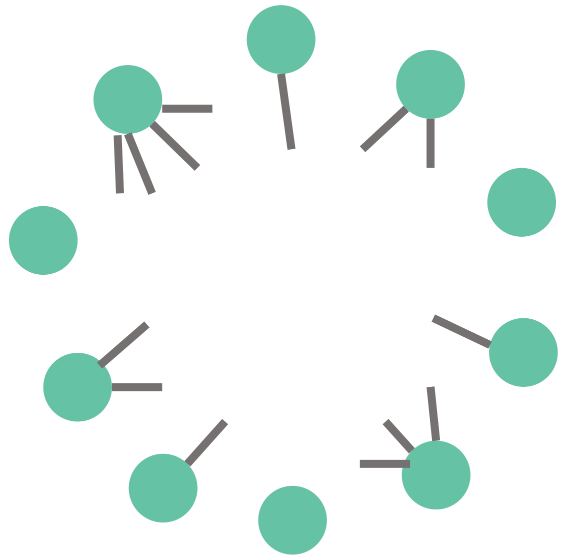
Erdos- Renyi

n nodes, every pair connects with probability p.



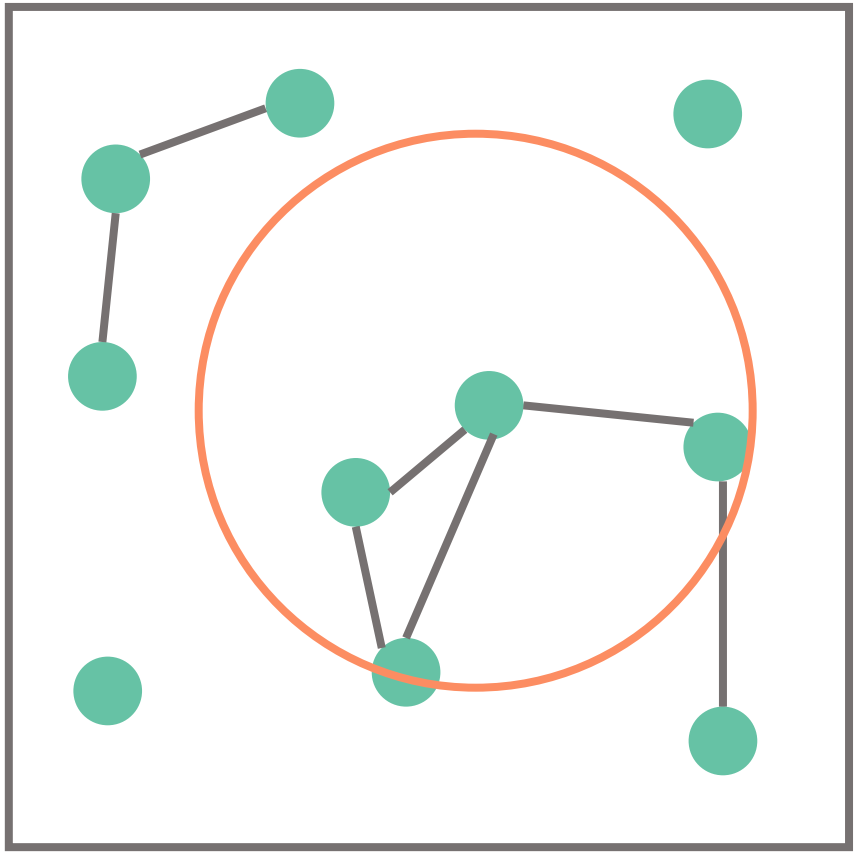
Configuration model

n nodes, with degrees $d_1, \ldots, d_n$. Connect 'stubs' randomly

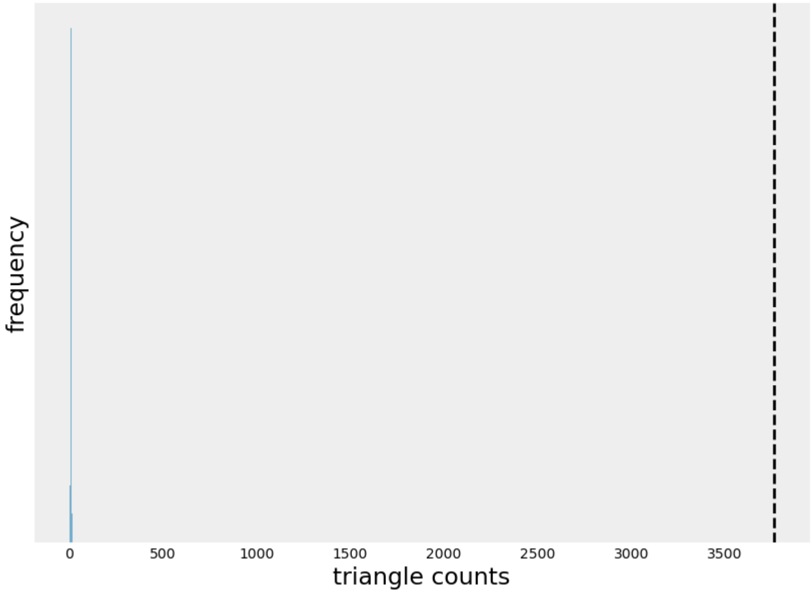


Geometric random graph

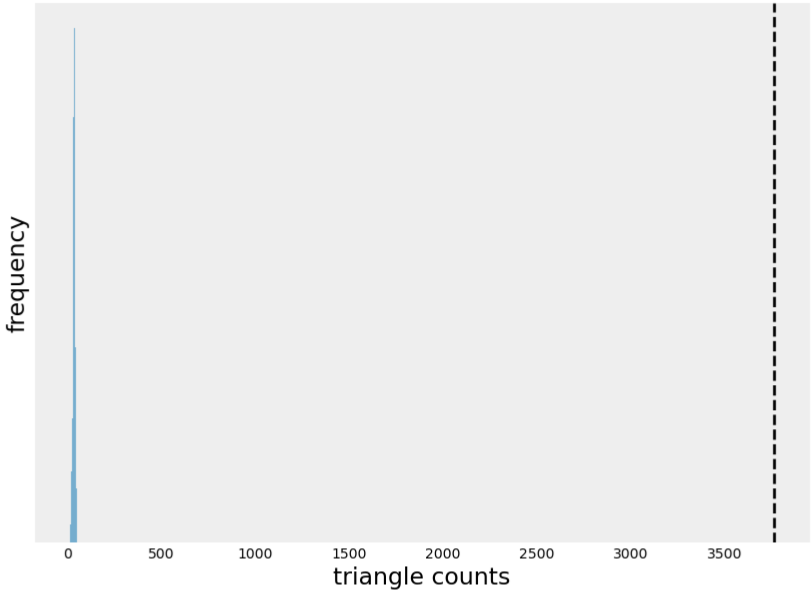
n nodes with uniform location in $[0,1]^2$ box. Connect all nodes within radius r.



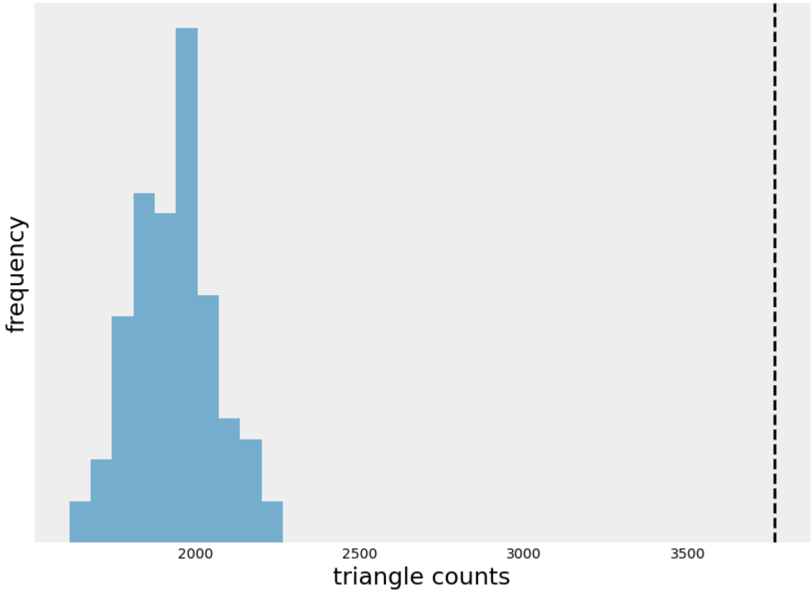
Different random graph models give different conclusions



Erdos-Renyi
`Z-score is 1365`

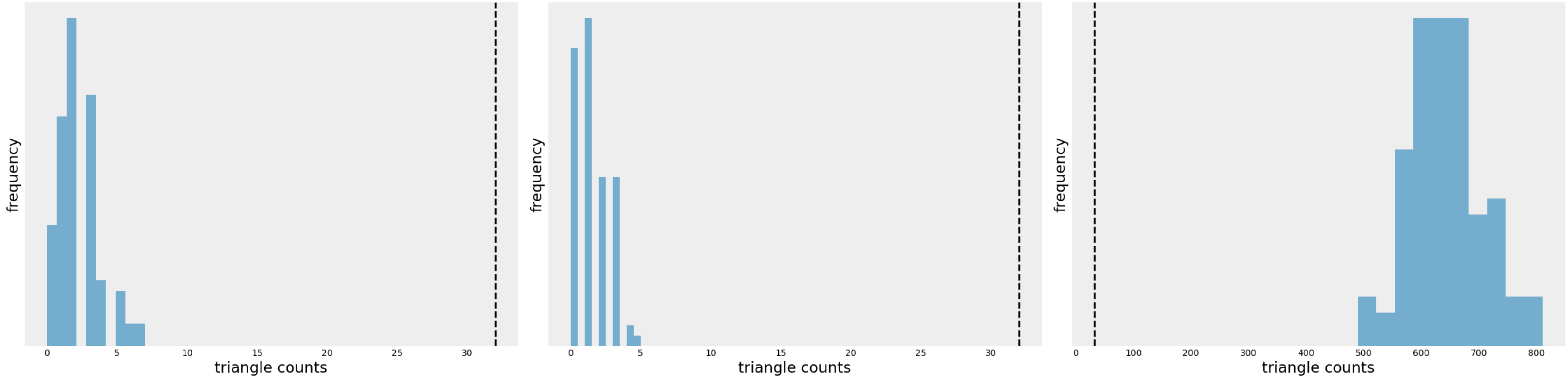


Configuration model
`Z-score is 555`



Geometric random graph
`Z-score is 14`

Different random graph models give different conclusions



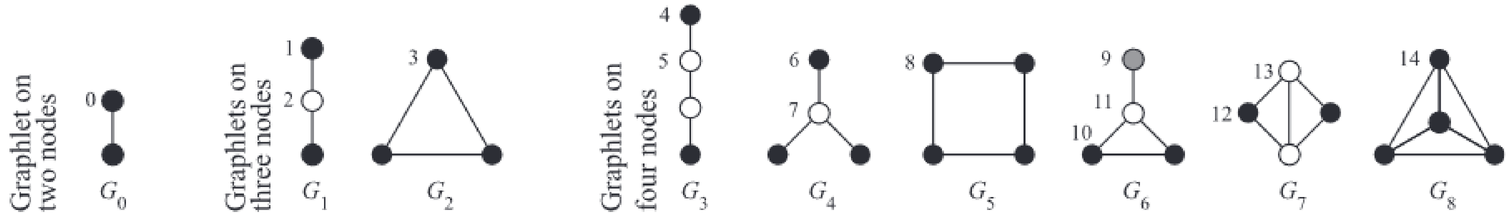
Erdos-Renyi
`Z-score is 20`

Configuration model
`Z-score is 26`

Geometric random graph
`Z-score is -10`

Conclusions

- Counting is often faster than listing



- Smart sampling techniques approximate counts in large networks