

Learning from Networks

Graphlets and Motifs

Fabio Vandin

November 7th, 2024

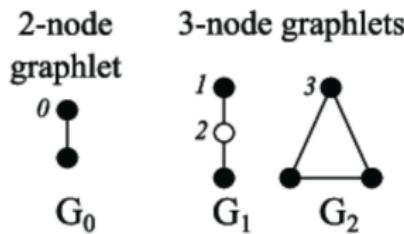
Graphlets

Other features that can be computed for a graph or for a node:
the counts (or other scores) of *graphlets*.

Definition

A **graphlet** is a small connected subgraph.

importante:
- dimensione (# nodi)



We are interested in counting the number of times a graphlet appears exactly as a subgraph.



appare 5 volte, ma con queste condizioni appare 2 volte

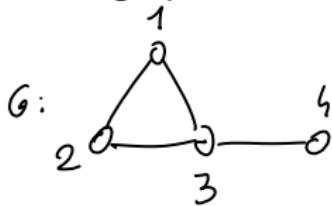
Graphlets (continue)

Definition

Given a graph $G = (V, E)$ let $S \subset V$. The *induced subgraph* $G[S] = (S, E')$ is the graph with:

- vertex set S ;
- edge set E' with $E' = \{(u, v) \in E : u \in S, v \in S\}$

What does it mean that a graphlet *appears* as an induced subgraph?

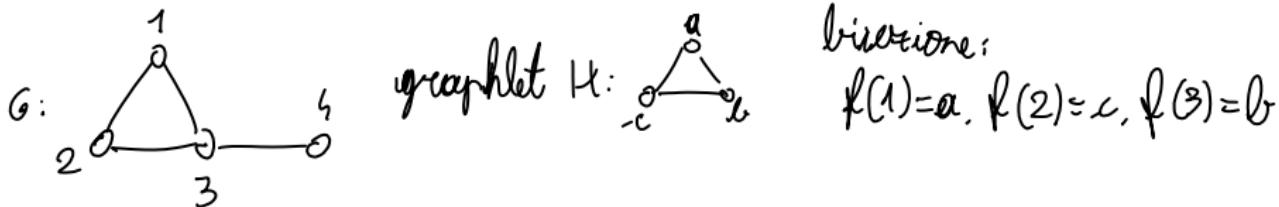


graphlet $H:$ \Rightarrow quante volte appare?
in teorema 6 (possibili mapping fra nodi di G e di H), ma possibile anche 1

Graphlets (continue)

Definition

Two graphs $G = (V_G, E_G)$ and $H = (V_H, E_H)$ are *isomorphic*, denotes as $G \simeq H$ if there exists a *bijection* $f : V_G \rightarrow V_H$ such that $(u, v) \in E_G$ if and only if $(f(u), f(v)) \in E_H$.



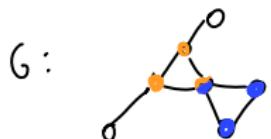
Informally: the count of a graphlet $H = (V_H, E_H)$ in a graph G is the number of subgraphs of G to which H is isomorphic to.

The count/score of a graphlet can be a feature at the graph level or at the node level.

Graphlets: Graph Level

Definition

Given a graphlet $H = (V_H, E_H)$ and a graph $G = (V, E)$, the count $c(H, G)$ of H in G is $c(H, G) = |\{S \subset V : H \simeq G[S]\}|$



$H:$  $c(H, G) = 2$

$H:$  $c(H, G) = 8$

Let's define the corresponding computational problem.

Definition

Given a graphlet $H = (V_H, E_H)$ and a graph $G = (V, E)$ the graphlet counting problem asks to compute $c(H, G)$.

Graphlet Counting: Computational Complexity

Let's define the **Subgraph Isomorphism Problem** (*decision*)

Definition

Given a graph H and a graph G , the subgraph isomorphism problem asks to determine whether G contains a subgraph isomorphic to H .

Is it an easy problem? Or a difficult one?

Proposition

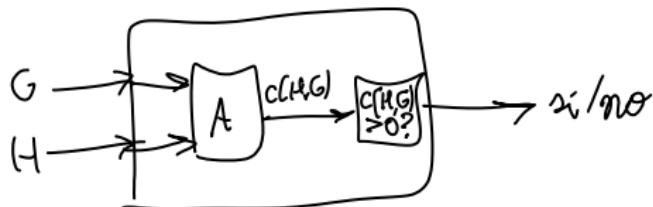
The subgraph isomorphism problem is NP-complete.

Graphlet Counting: Computational Complexity (continue)

Proposition

A polynomial time algorithm for the graphlet counting problem implies a polynomial time algorithm for the subgraph isomorphism problem.

Poniamo dg. A poly-time per graphlet counting \Rightarrow
 \Rightarrow posso estrarre dg. B per subgraph isomorphism



Graphlet Counting: Algorithms

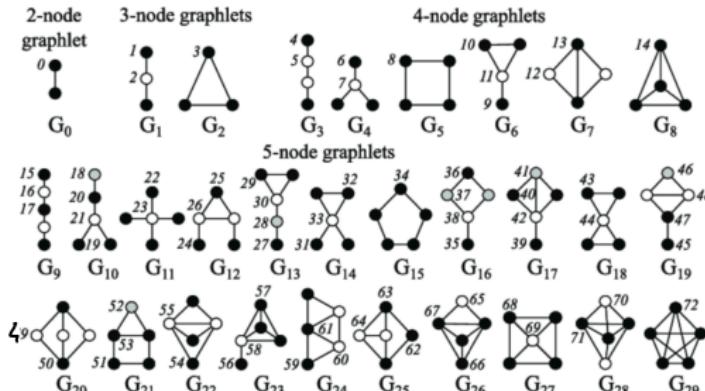
Several algorithms have been developed for the graphlet counting problem.

In practice: often want to solve the following problem

Input: graph G , integer $k \in \mathbb{N}^+$

Output: the counts in G of all graphlets with k nodes

Note: number of graphlets with k nodes increases rapidly with k



numeri sui nodi per sapere quante volte contare

Graphlet Counting: Naïve Algorithm

In: graph $G = (V, E)$, $k \in \mathbb{N}^+$

Out: $c(H, G)$ # graphlets H con k nodi

1 - $\forall H : c(H, G) = 0$

2 - $\forall S \subseteq V, |S| = k$ do:

$\forall H$ di dim k do:

if $H \cong G[S]$ then $c(H, G) += 1$

3 - return $c(H, G)$ # H ;
 subgraph isomorphism
 può essere risolto
 altrettanto efficientemente
 per k piccoli

Complexità: $\mathcal{O}\left(\binom{n}{k}\right) \in \mathcal{O}(n^k)$, $n = |V|$

ESU Algorithm

ESU (Exact Subgraph Enumeration) algorithm presented in:
Wernicke, S. (2006) *Efficient detection of network motifs.*
IEEE/ACM Transactions on Computational Biology and
Bioinformatics.

Input: graph G , integer $k \in \mathbb{N}^+$

Output: the counts in G of all graphlets with k vertices

Works in two phases:

- first phase: enumerate all connected subgraphs with k nodes in G
- second phase: compute the count $c(H, G)$ of each graphlet H with k vertices

An Approximation Algorithm

We are now going to describe an approach to approximate the count of graphlets in a graph

Based on **color-coding** technique: to count subgraphs of size k , randomly color the vertices with k colors, and count only colorful subgraphs

Colorful subgraph: every vertex has a different color

Color-coding technique: introduced in “Alon, Noga, Raphael Yuster, and Uri Zwick. Color-coding. Journal of the ACM (JACM) 42.4 (1995): 844-856.”

First used for graphlet counting in “Alon, N., Dao, P., Hajirasouliha, I., Hormozdiari, F., Sahinalp, S. C. (2008). Biomolecular network motif counting and discovery by color coding. Bioinformatics, 24(13), i241-i249.”

Color-Coding for Subgraph Counting



: induced count: 0

: non-induced count: 3

For simplicity:

- count the number of non induced occurrences of H in G
- we present the version of the algorithm that approximate the count of given graphlet H
- we look at the details only when the graphlet is a path of length k

$\Downarrow H: \circ-\circ-\circ-\circ$

Color-Coding for Subgraph Counting

Algorithm ColorCodingCount(G, H, t)

Input: graph $G = (V, E)$ with $|V| = n$ and $|E| = m$; graphlet H with k nodes; $t \in \mathbb{N}^+$

↓
percorso
—o—o

Output: estimated count of \overbrace{H} in G *non induced*

```
count ← 0;  
forall  $i \leftarrow 1$  to  $t$  do  
     $G' \leftarrow \text{RandomColor}(G, k);$   
     $tmp \leftarrow \text{CountColorful}(G', H);$   
    count ← count + tmp;  
  
return  $\frac{1}{t} count \frac{k^k}{k!};$ 
```

Color-Coding for Subgraph Counting

Algorithm RandomColor(G, k)

Input: graph $G = (V, E)$ with $|V| = n$ and $|E| = m$; $k \in \mathbb{N}^+$

Output: graph G' with each vertex having a color

$G' \leftarrow (V, E);$

forall $v \in V$ **do**

 color v in G' with a value (color) chosen uniformly at

 random in $\{1, \dots, k\}$;

return G' ;

Algorithm CountColorful(G' , H)

coloratio

Input: graph $G = (V, E)$ with $|V| = n$ and $|E| = m$; graphlet H with k nodes

Output: number of colorful occurrences of H in G

The algorithm is based on a dynamic programming approach.

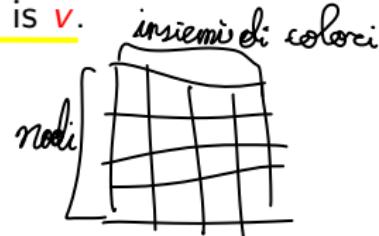
We consider the approach when H is a path (with k vertices)

For each vertex v :

- $\text{col}(v)$ = the color of v
- for each subset $S \subseteq \{1, \dots, k\}$: $c(v, S)$ = number of colorful paths (with colors S) for which one endpoint is v .

Let $\ell \in \{1, \dots, k\}$:

$$c(v, \ell) = \begin{cases} 1 & \text{if } \text{col}(v) = \ell \\ 0 & \text{otherwise} \end{cases}$$



For each vertex v and color set $S \subseteq \{1, \dots, k\}$ with $|S| > 1$:

$$c(v, S) = \begin{cases} \sum_{u: (u,v) \in E} c(u, S \setminus \{\text{col}(v)\}) & \text{if } \text{col}(v) \in S \\ 0 & \text{otherwise} \end{cases}$$

Then the number of colorful paths of length k is

$$\frac{1}{2} \sum_{v \in V} c(v, \{1, \dots, k\})$$

Proposition

Let $\tilde{c}(H, G)$ be the output of $\text{ColorCodingCount}(G, H, t)$, and $\varepsilon, \delta \in (0, 1)$

If $t \geq \frac{1}{\varepsilon^2} \frac{1}{\delta} \frac{k^k}{k!}$ then with probability $\geq 1 - \delta$ we have

$$\tilde{c}(H, G) \in [(1 - \varepsilon)c(H, G), (1 + \varepsilon)c(H, G)].$$

Dim: basata su Chernoff bound per var[$\sum p_i$] \Rightarrow
 $\Rightarrow \text{Var}[\varepsilon Z_j] \Rightarrow Z_j$ non ind. \Rightarrow bound $\text{Cor}[Z_i, Z_k]$

migliori bound: $t \geq \frac{1}{\varepsilon^2} \frac{k^k}{k!} \log \frac{1}{\delta}$

Analysis: Time Complexity

Alt:

- $\text{RandomGraph}(G, k)$: $\Theta(n)$, $n = |V|$
- $\text{CountColorful}(G, t)$: $\forall \text{set } S \subseteq \{1, \dots, k\}$,
calcolo $\sum_{v \in V} c(u, S \setminus \{\text{col}(v)\})$, $\forall v \Rightarrow$
già calcolato

$$\Rightarrow \Theta(\sum_{v \in V} \deg(v)) \in \Theta(m) \quad \text{se } G \text{ è connesso}$$

$$\text{complex. totale: } \Theta(2^k m)$$

$$\text{Totale per it: } \Theta(2^k m) \Rightarrow \text{totale: } \Theta(t 2^k m)$$

lower bound dato in Proc: (=)

$$\Theta\left(2^k m \cdot \frac{1}{e^2} \frac{1}{t} \frac{k^k}{k!}\right) \Rightarrow \text{dato che } K! \geq \left(\frac{k}{e}\right)^k,$$

(Stirling)

$$\Theta\left(\frac{1}{e^2} \frac{1}{t} \cdot e^k 2^k m\right) \Rightarrow \text{sempre } \exp(k), \text{ ma non è } n$$

$$E\left[\frac{1}{t} \text{count}_{\frac{k^k}{k!}}\right] = c(H, G)$$

Dim: $X_i = \text{val. di temp. in it. } i \}$
 $Y = \frac{1}{t} \text{ count}_{\frac{k^k}{k!}}$ } $\forall v$.

$$Y = \frac{1}{t} \left(\sum_{i=1}^t X_i \right) \frac{k^k}{k!} \Rightarrow E[Y] = \frac{1}{t} \frac{k^k}{k!} + E[X_i]$$

ordiniamo non-individ. occ. di H in G :

$$j = 1, 2, \dots < c(H, G)$$

$$Z_j = \begin{cases} 1 & \text{se occ. } j \text{ di } H \text{ è colorata in it. } i \\ 0 & \text{else} \end{cases}$$

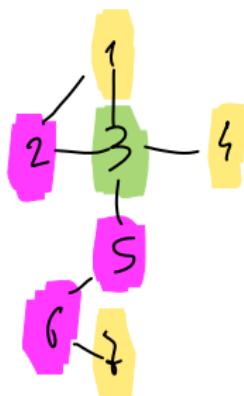
$$X_i = \sum_{j=1}^{c(H, G)} Z_j \Rightarrow E[X_i] = \sum_{j=1}^{c(H, G)} E[Z_j]$$

$$E[Z_j] = \Pr[Z_j = 1] = \frac{\# \text{colorful}}{\# \text{colorings}} = \frac{k!}{t! \cdot \text{colorings}} = \frac{k!}{k^k}$$

$$E[X_i] = c(H, G) \frac{k!}{k^k} \Rightarrow E[Y] = \frac{1}{t} \frac{k^k}{k!} + c(H, G) \frac{k!}{k^k} =$$

$$= c(H, G)$$

G



H: $\circ - \circ - \circ$ ($k=3$)

6 occorrenze colorate

v	1	2	3	4	5	6	7
1	1						3
2		1	1	1	1		3
3			1		2	2	2
4	1					1	2
5		1			1		2
6		1		1			
7	1			1			

$c(1, \text{yellow}) =$
 $= c(2, \text{pink}) + c(3, \text{green}) = 1$

totale: 6

$l \in \{1, \dots, k\}$

$$c(v, l) = \begin{cases} 1 & \text{se } \text{col}(v) = l \\ 0 & \text{else} \end{cases}$$

$$c(v, l) = \begin{cases} \sum_{u \in N_v} c(u, l \setminus \{\text{col}(u)\}) & \text{if } \text{col}(u) \neq l \\ 0 & \text{else} \end{cases}$$

Color-Coding: Improved Algorithms for Subgraph Counting

All *non-induced* graphlets with small treewidth

- Alon, N., Dao, P., Hajirasouliha, I., Hormozdiari, F., Sahinalp, S. C. Bioinformatics, 2008
- Chakaravarthy, V. T., Kapralov, M., Murali, P., Petrini, F., Que, X., Sabharwal, Y., Schieber, B. IEEE International Parallel and Distributed Processing Symposium, 2016

Larger (7-9 vertices) *induced* graphlets in large graphs:

- M. Bressan, S. Leucci, and A. Panconesi. Proc. VLDB Endowment, 2019

Experimental Evaluation

M. Bressan, S. Leucci, A. Panconesi. MOTIVO: fast motif counting via succinct color coding and adaptive sampling. Proc. VLDB Endow., 2019

"We ran all our experiments on a commodity machine equipped with 64GB of main memory and 48 Intel Xeon E5-2650v4 cores at 2.5GHz with 30MB of L3 cache. We allocated 880GB of secondary storage on a Samsung SSD850 solid-state drive, dedicated to the treelet count tables of MOTIVO."

Table 1: our graphs (* = with biased coloring)

graph	M nodes	M edges	source	k
FACEBOOK	0.1	0.8	MPI-SWS	9
BERKSTAN	0.7	6.6	SNAP	9
AMAZON	0.7	3.5	SNAP	9
DBLP	0.9	3.4	SNAP	9
ORKUT	3.1	117.2	MPI-SWS	7
LIVEJOURNAL	5.4	49.5	LAW	8
YELP	7.2	26.1	YLP	8
TWITTER	41.7	1202.5	LAW	6 (7*)
FRIENDSTER	65.6	1806.1	SNAP	6 (7*)

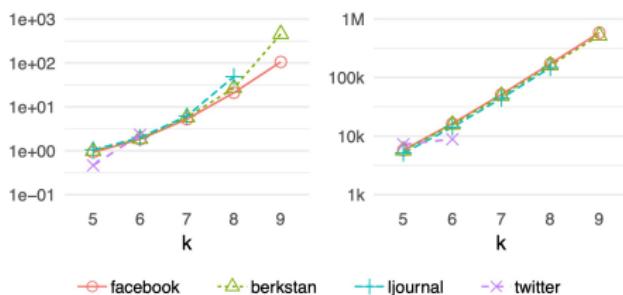


Figure 6: MOTIVO's build-up time (seconds per million edge) and space usage (bits per input node).

Motifs

Definition

A ***motif*** for a graph G is a **graphlet** that is *significantly overrepresented* in G .

How do we measure if a graphlet is significantly overrepresented?

Comparison with appropriate random graphs:

- large z -score
- small p -value

Motifs (continue)

Sometimes the normalized z-scores of all graphlets of size up to k are used as graph-level features.

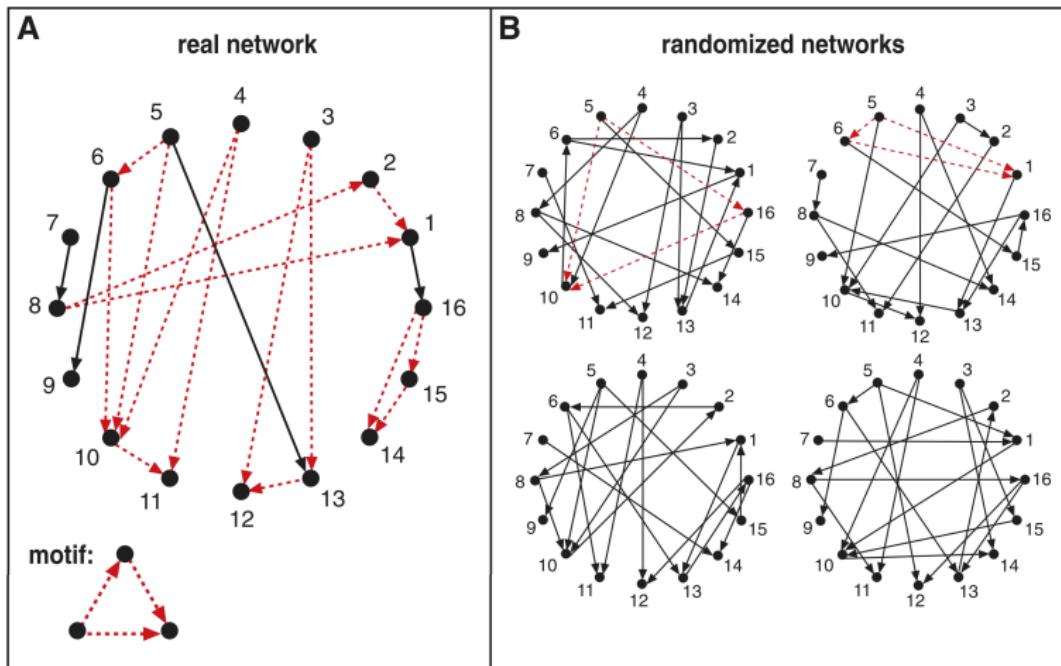
Definition

Given a graph G and $k \in \mathbb{N}^+$, the network significance profile of G is the vector of the normalized z-scores of the graphlets of size $\leq k$ in G . If $\mathcal{G}(k)$ is the set of graphlets of size $\leq k$ and $z_G(H)$ is the z-score for graphlet H in G , then the normalized z-score of H in G is

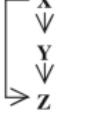
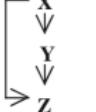
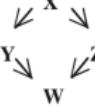
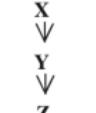
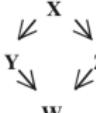
$$\frac{z_G(H)}{\sum_{H' \in \mathcal{G}(k)} z_G(H')}$$

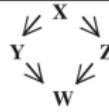
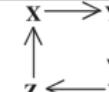
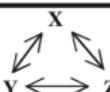
Motifs: Example

Milo et al. (2002) *Network Motifs: Simple Building Blocks of Complex Networks*. Science.

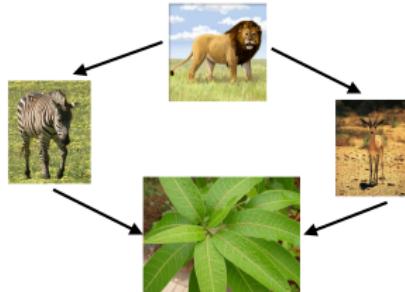
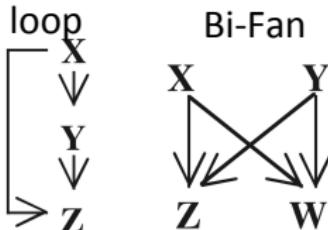


Random graph model: each node has same indegree and same outdegree.

Network	Nodes	Edges	N_{real}	$N_{\text{rand}} \pm \text{SD}$	Z score	N_{real}	$N_{\text{rand}} \pm \text{SD}$	Z score	N_{real}	$N_{\text{rand}} \pm \text{SD}$	Z score
Gene regulation (transcription)				Feed-forward loop			Bi-fan				
<i>E. coli</i>	424	519	40	7 ± 3	10	203	47 ± 12	13			
<i>S. cerevisiae*</i>	685	1,052	70	11 ± 4	14	1812	300 ± 40	41			
Neurons				Feed-forward loop			Bi-fan			Bi-parallel	
<i>C. elegans†</i>	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
s38584	20,717	34,204	413	10 ± 3	120	1739	6 ± 2	800
s38417	23,843	33,661	612	3 ± 2	400	2404	1 ± 1	2550
s9234	5,844	8,197	211	2 ± 1	140	754	1 ± 1	1050
s13207	8,651	11,831	403	2 ± 1	225	4445	1 ± 1	4950
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
s420	252	399	20	1 ± 1	18	10	1 ± 1	10
s838‡	512	819	40	1 ± 1	38	22	1 ± 1	20
World Wide Web				Feedback with two mutual dyads		Fully connected triad		Up-linked mutual dyad
nd.edu§	325,729	1.46e6	1.1e5	$2e3 \pm 1e2$	800	6.8e6	$5e4 \pm 4e2$	15,000
							1.2e6	$1e4 \pm 2e2$
								5000

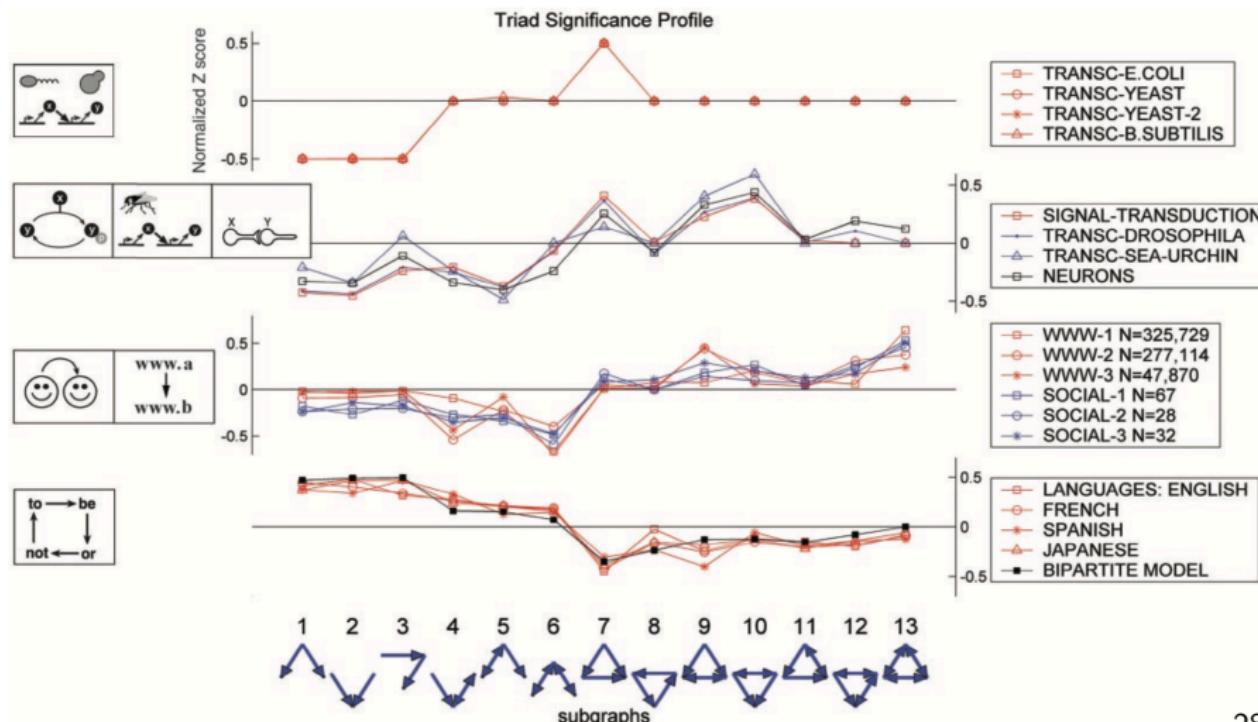
Feed-forward

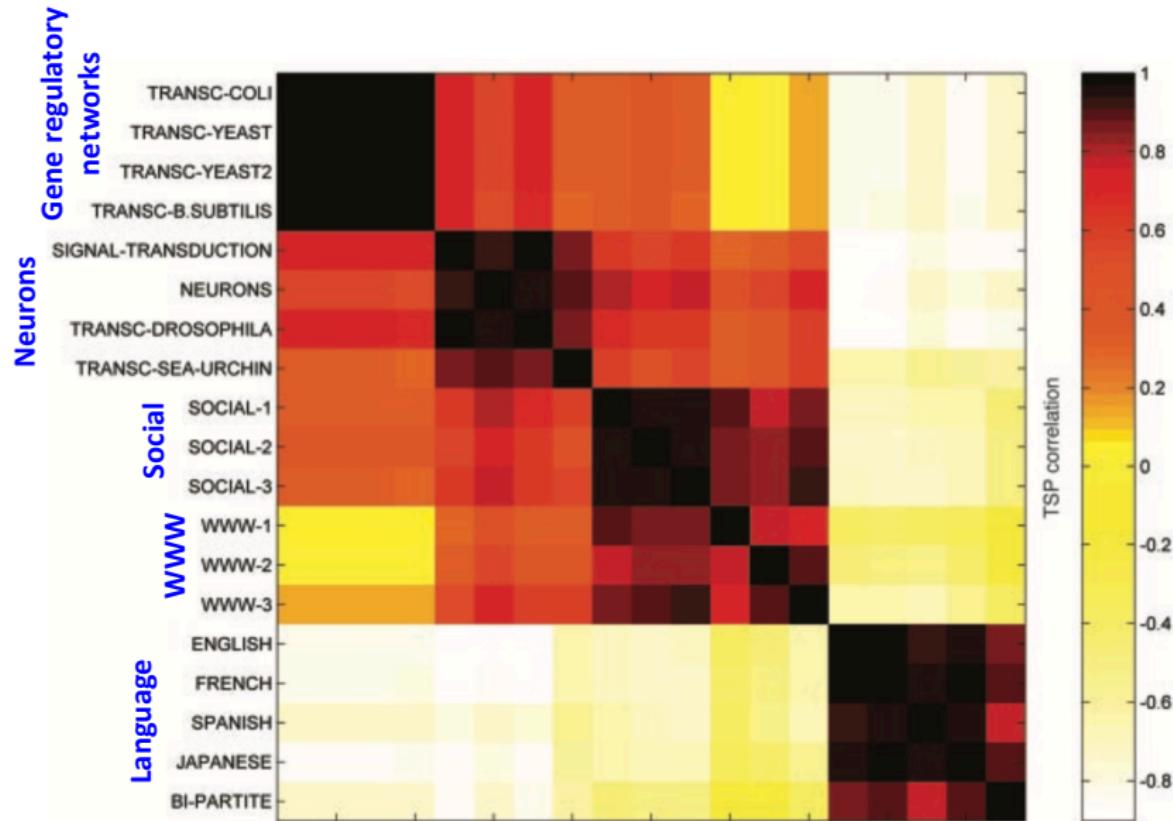


- networks of neurons and gene networks contain similar motifs:
 - feed-forward loops and bi-fan structures
 - both are information processing networks with sensory and acting components
- food webs have parallel loops: prey of a particular predator share prey
- WWW network has bidirectional links: design that allows the shortest path between sets of related pages

Motifs: Example

Milo et al. (2004) *Superfamilies of Evolved and Designed Networks*. Science.



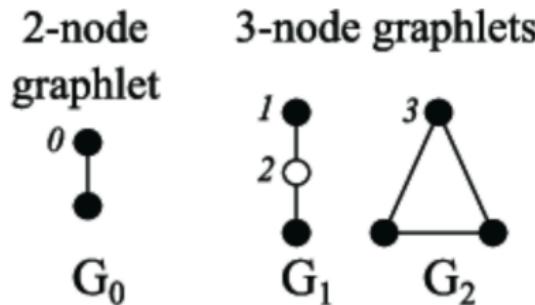


Graphlets and Motifs: Node Level

Graphlets and motifs can be used to define node level features as well.

Definition

Given a graphlet H , its **orbits** are the different positions in which a node can appear (up to automorphism).



Graphlets and Motifs: Node Level (continue)

Definition

Given a graph $G = (V, E)$, $k \in \mathbb{N}^+$, and a node $v \in V$, the *graphlet degree vector* $GDV(v, G)$ of v in G is the vector where each component is the number of occurrences of v in a given orbit of a given graphlet of size $\leq k$.

One can define a similar vector based on *motifs* (i.e., z -scores) instead.

Algorithm

How can we obtain the $GDV(v, G)$ for a node v ?

Simple algorithm:

- ① based on the graphlet size k , compute the maximum distance r between v and a node in a graphlet H of size $\leq k$
- ② collect the subgraph G' of G containing v and all its neighbours up to distance r
- ③ use the first phase of the ESU algorithm to enumerate all appearances of graphlets in G'
- ④ compute $GDV(v, G)$

Extensions

There are several extensions of the problems we discussed, for example

- Graphlet/motif definition:
 - directed
 - node colors
 - temporal networks
 - ...
- Variants of graphlet scores:
 - different concepts of *appearing as a subgraph*
 - different measures for significance
 - ...
- ...

Open Problems

There are several open problems that are the focus of recent research:

- efficient sampling algorithms with guarantees for graphlets
- efficient algorithms to compute p -values for motifs
- efficient algorithms for large values of k
- ...