A Faster Algorithm for Detecting Network Motifs

Sebastian Wernicke

Institut für Informatik, Friedrich-Schiller-Universität Jena,
Ernst-Abbe-Platz 2, D-07743 Jena, Fed. Rep. of Germany
wernicke@minet.uni-jena.de

Outline

2. 3.

A Faster Algorithm

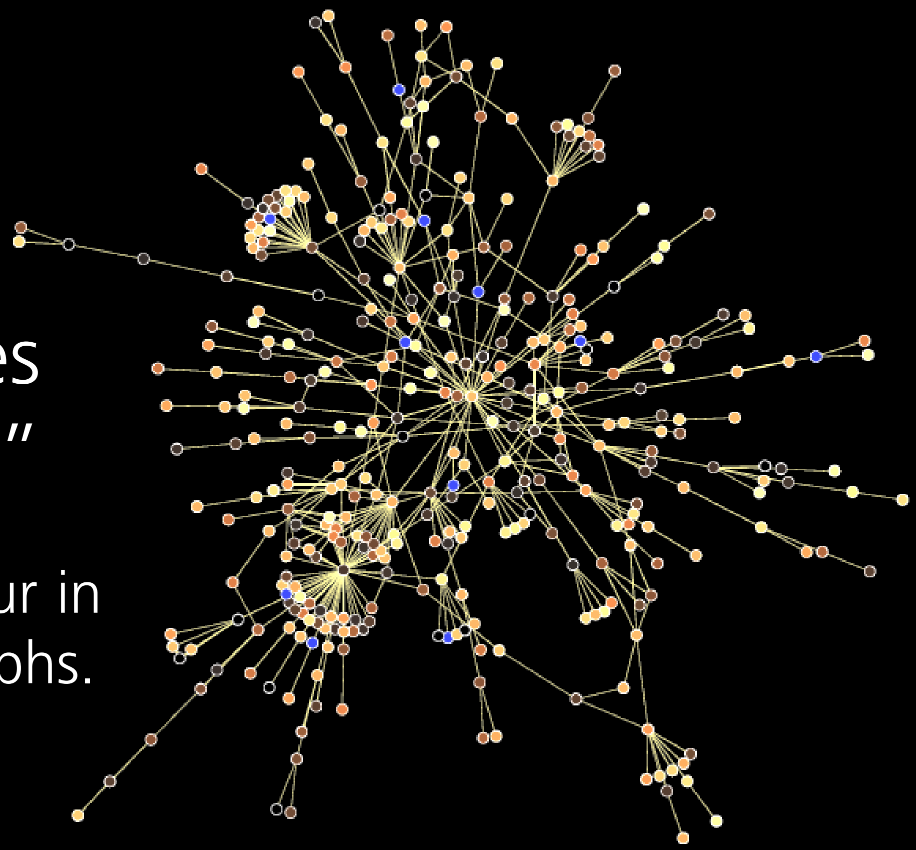
for Detecting Network Motifs

1.

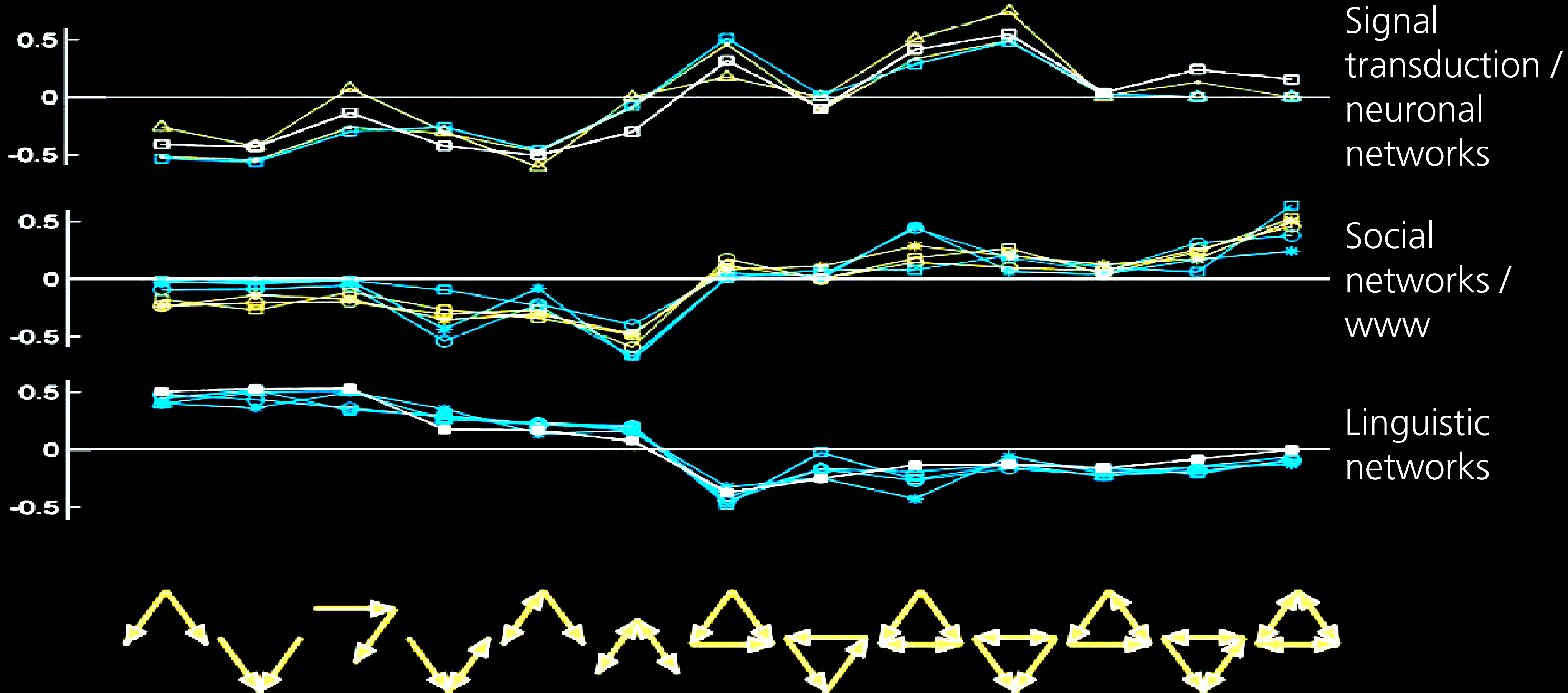
Network Motifs

"Evolution preserves modules that define specific function."

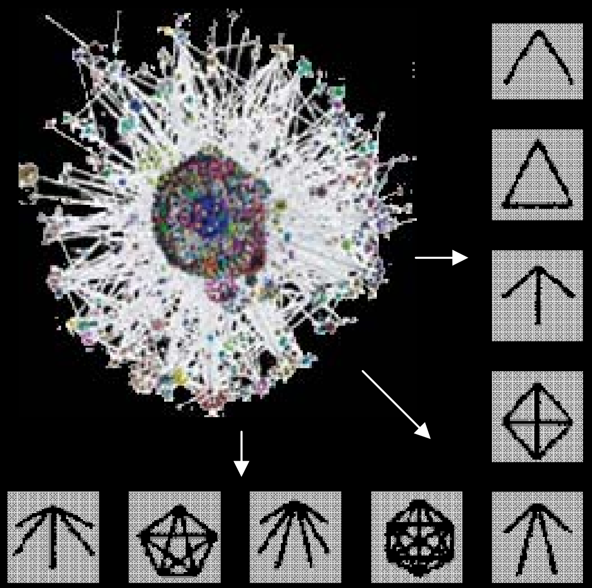
Motifs are those subgraphs which occur in higher frequencies than in random graphs.



Network Motifs

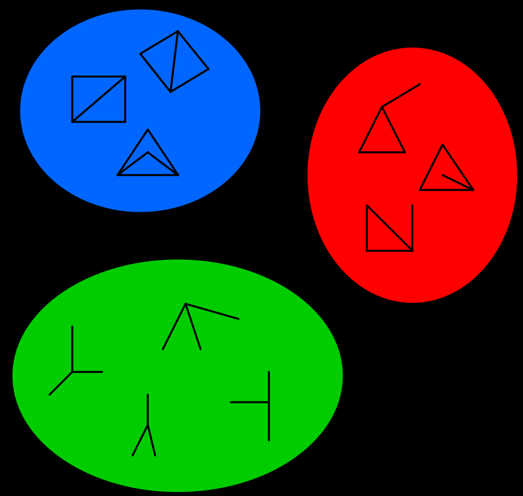


Detecting Motifs



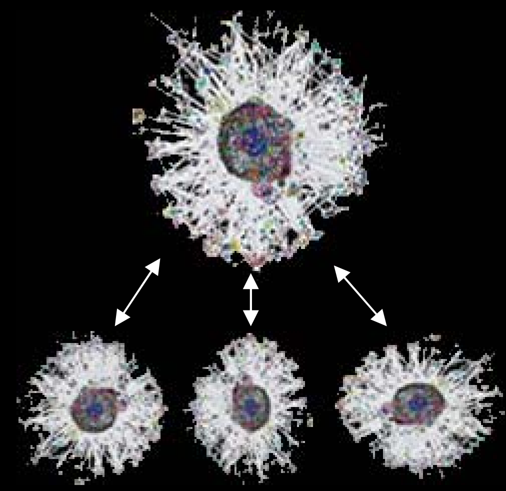
Count

HARD



Group

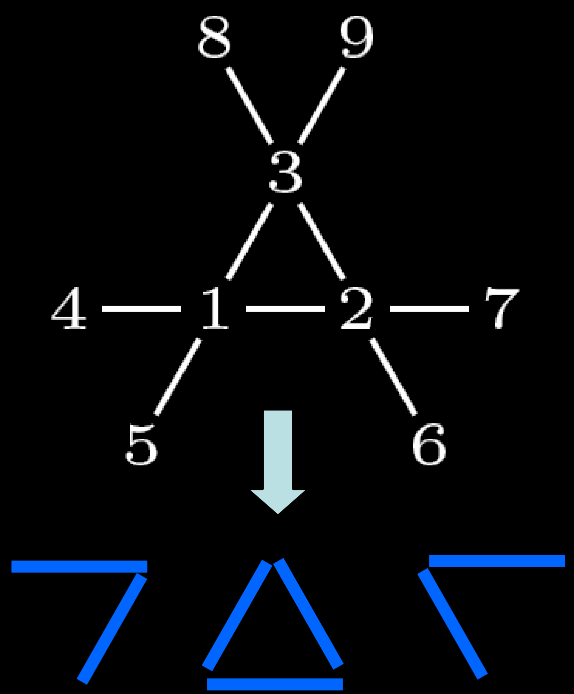
HARD



Compare

HARD

Subgraph Sampling



Algorithm: EDGE SAMPLING(G, k) (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.

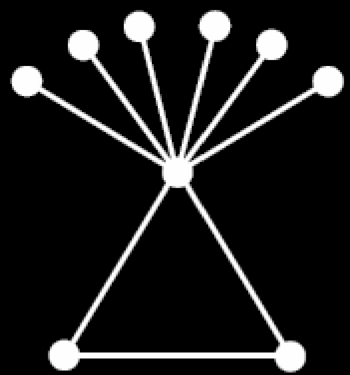
$\{u, v\} \leftarrow$ random edge from E
$V' \leftarrow \{u, v\}$
while $|V'| \neq k$ **do**
    $\{u, v\} \leftarrow$ random edge from $V' \times N(V')$
    $V' \leftarrow V' \cup \{u\} \cup \{v\}$
return V'

Previous Approach [Kashtan et al., *Bioinformatics*, 2004]

Subgraph Sampling

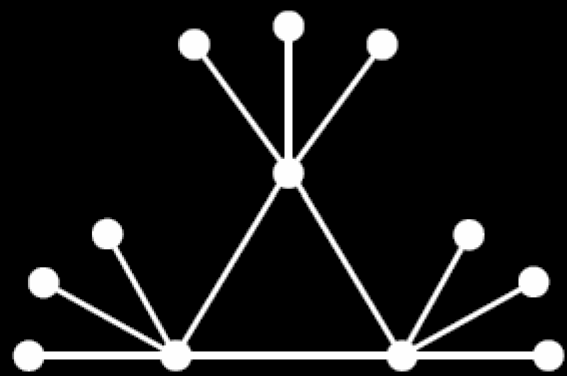
Both graphs have 28 size-3 subgraphs.

G_1



Probability of sampling the triangle here: 1/6

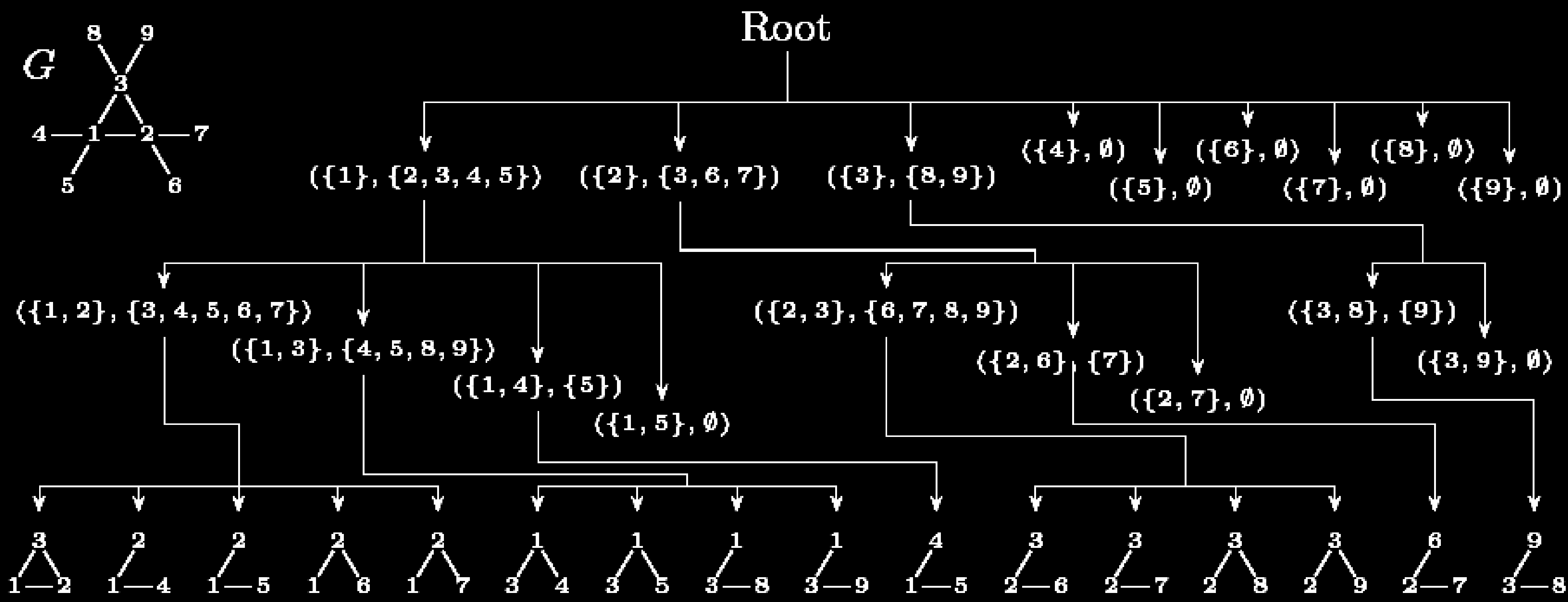
G_2



Probability of sampling the triangle here: 1/16

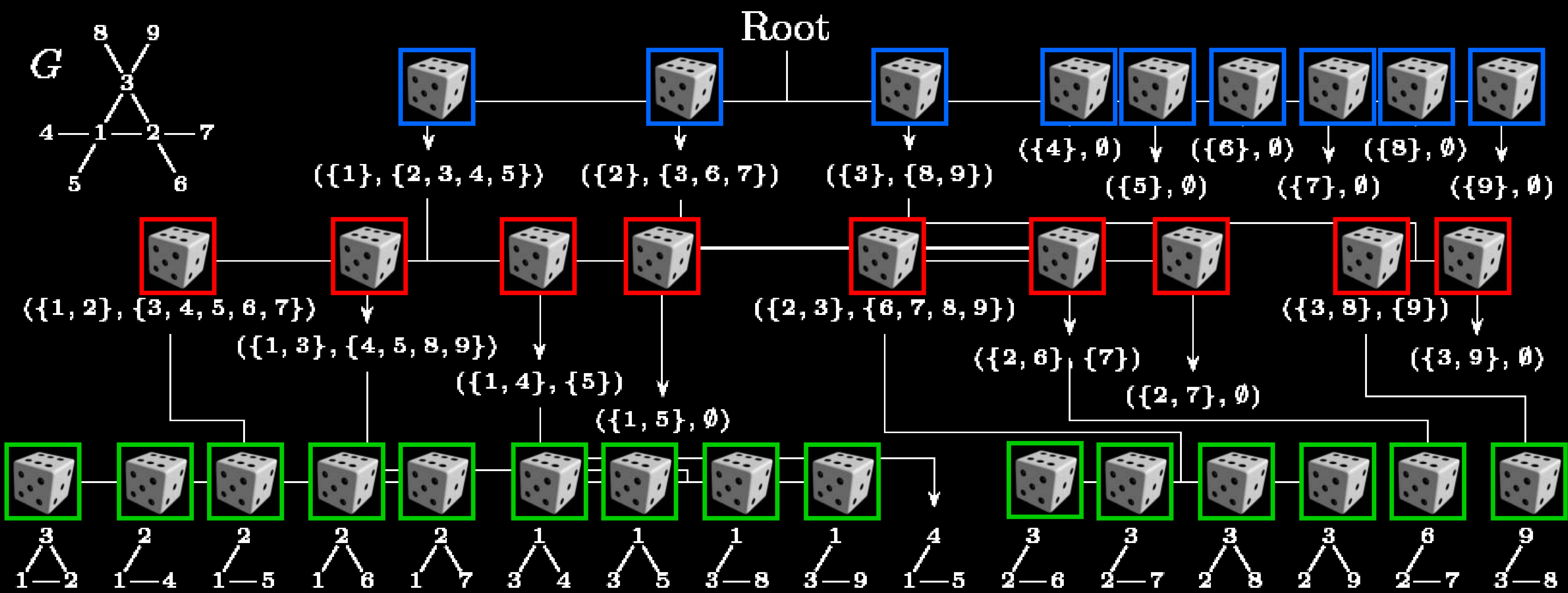
Previous Approach [Kashtan et al., *Bioinformatics*, 2004]

Subgraph Counting



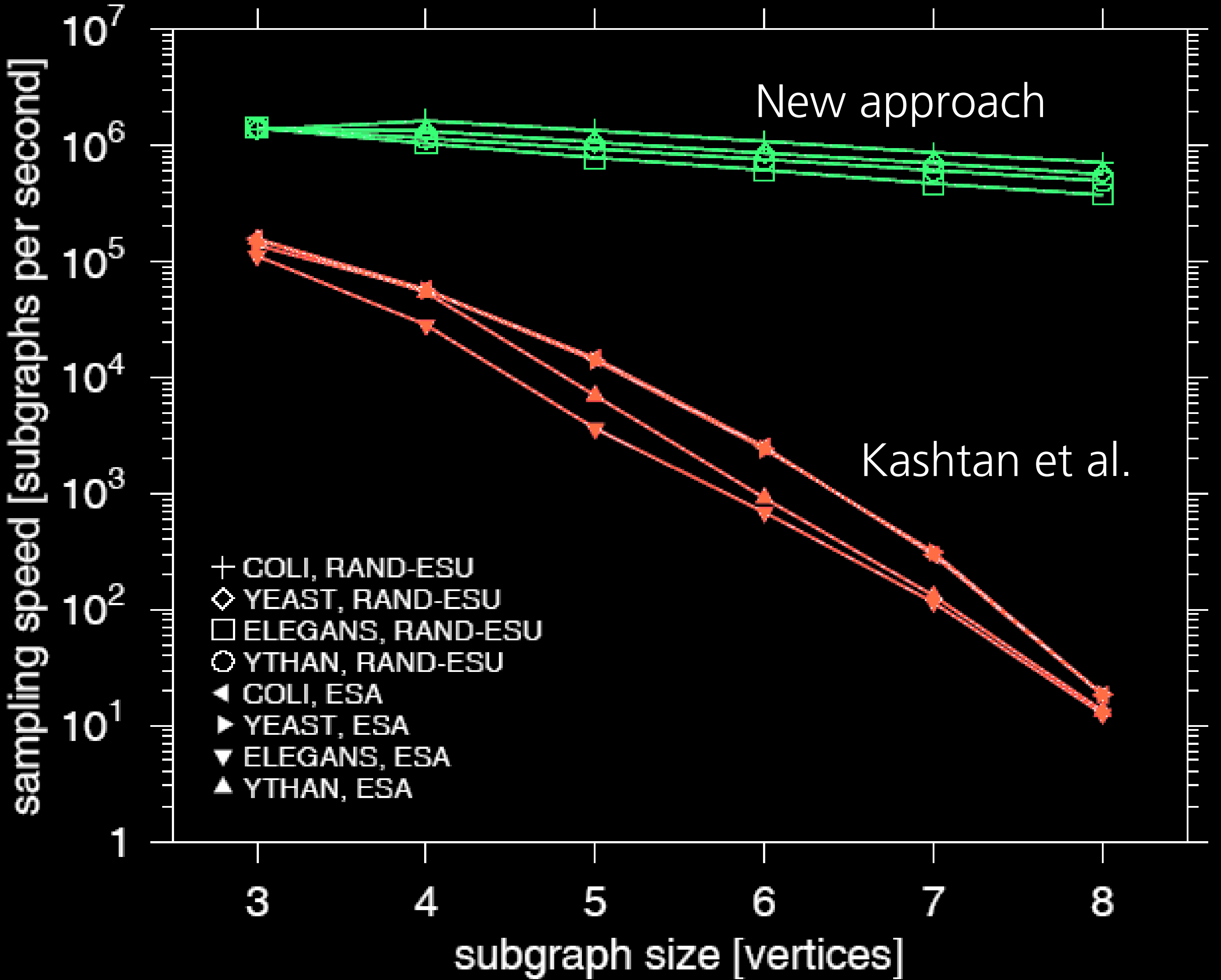
New approach is based on
deterministic subgraph enumeration.

Sampling from Counting

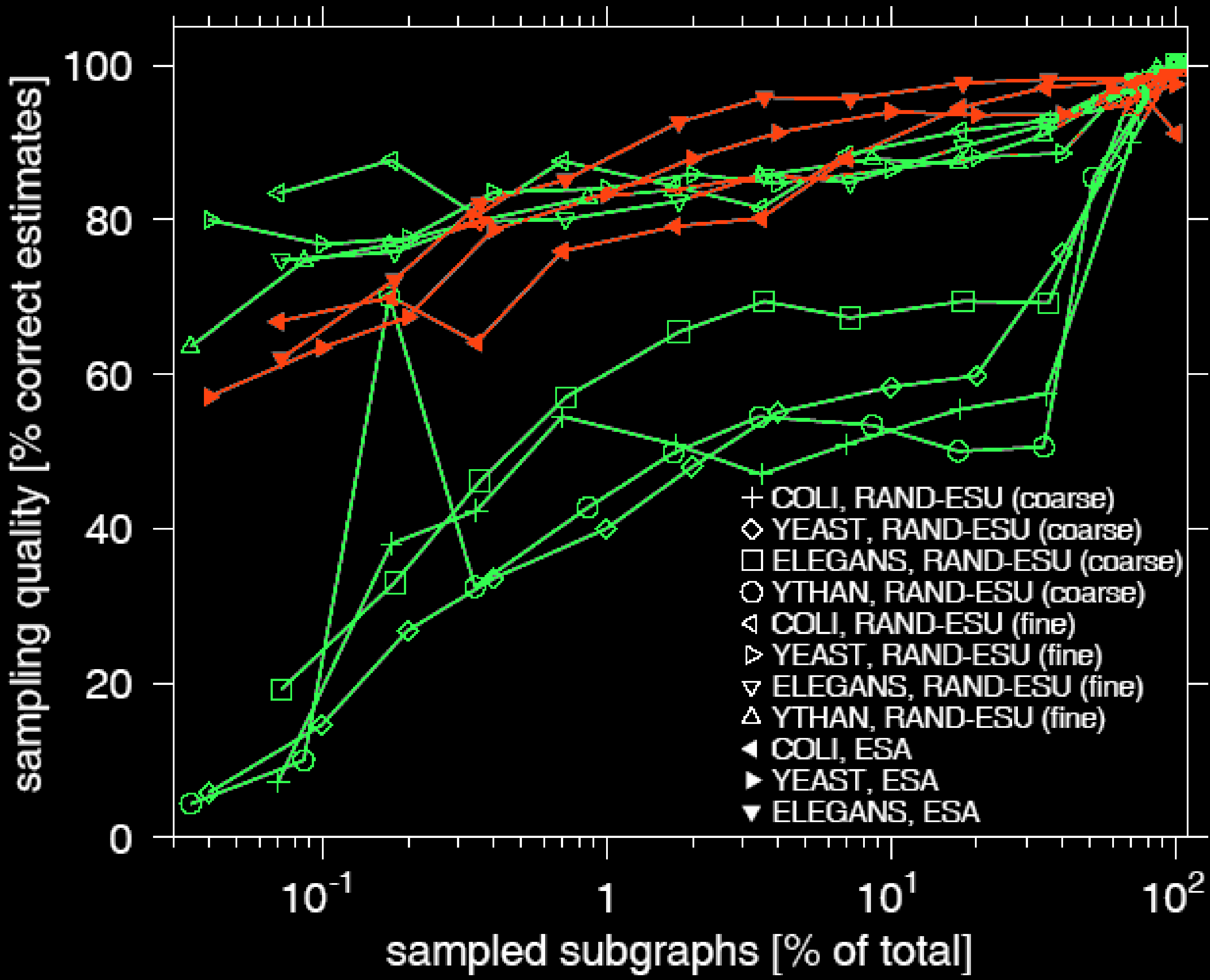


Randomized traversal of search tree yields uniform sampling.

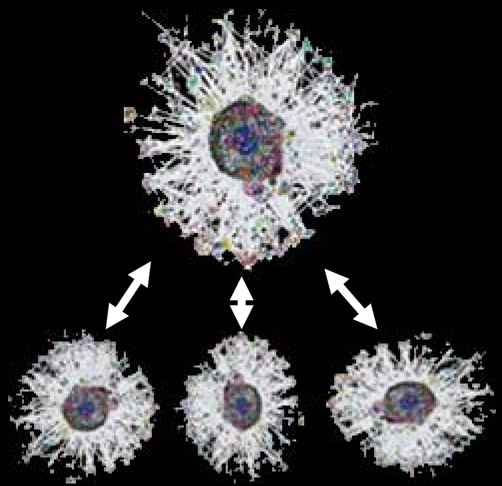
Performance

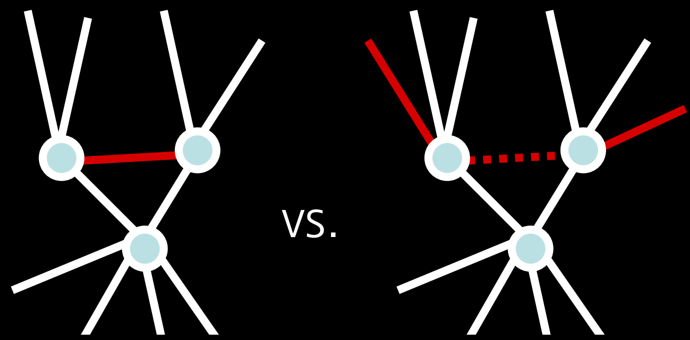


Quality


100
80
60
40
20
0
sampling quality [% correct estimates]
10^{-1}
1
10^{1}
10^{2}
sampled subgraphs [% of total]
COLI, RAND-ESU (coarse)
YEAST, RAND-ESU (coarse)
ELEGANS, RAND-ESU (coarse)
YTHAN, RAND-ESU (coarse)
COLI, RAND-ESU (fine)
YEAST, RAND-ESU (fine)
ELEGANS, RAND-ESU (fine)
YTHAN, RAND-ESU (fine)
COLI, ESA
YEAST, ESA
ELEGANS, ESA

Subgraph Significance





$$G(M,r) \sim \frac{\sqrt{2}(f/e)^{(f/2)}}{\exp(a^2+a+b)\cdot\prod_i(r_i!)}$$

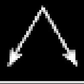
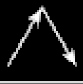
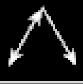
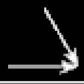
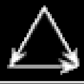
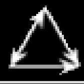
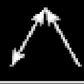
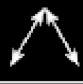
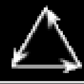
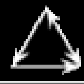
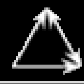
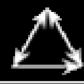
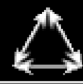
Old way: Explicit generation

> 1000 graphs with sampling and grouping for each (again)

New way: Just count!

Given *k* vertices: how often do they induce a given subgraph?

Direct Calculation Quality

														
COLI	$\langle \mathcal{C}_k^i \rangle$	9.1e-1	3.7e-2	1.9e-4	5.0e-2	1.4e-3	2.1e-6	7.6e-8	3.4e-7	2.9e-6	2.9e-5	8.0e-7	–	–
	$\langle \hat{\mathcal{C}}_k^i \rangle$	9.0e-1	4.2e-2	2.6e-4	5.5e-2	1.4e-3	2.1e-6	1.3e-7	8.7e-8	2.3e-6	4.4e-5	1.1e-7	8e-12	6e-15
$\langle \mathcal{C}_k^i \rangle / \langle \hat{\mathcal{C}}_k^i \rangle$		1.0	0.9	0.7	0.9	1.0	1.0	0.6	3.9	1.3	0.7	7.4	–	–
YEAST	$\langle \mathcal{C}_k^i \rangle$	9.1e-1	3.7e-2	1.8e-4	5.0e-2	1.4e-3	9.5e-7	–	2.6e-7	2.3e-6	2.9e-5	3.4e-7	–	–
	$\langle \hat{\mathcal{C}}_k^i \rangle$	8.9e-1	3.0e-2	1.2e-4	7.6e-2	1.2e-3	1.5e-6	2.8e-8	4.4e-8	5.4e-7	1.0e-5	1.0e-7	1e-14	1e-15
$\langle \mathcal{C}_k^i \rangle / \langle \hat{\mathcal{C}}_k^i \rangle$		1.0	1.2	1.5	0.6	1.2	0.7	–	6.1	4.3	2.9	3.3	–	–
ELEG.	$\langle \mathcal{C}_k^i \rangle$	2.0e-1	3.3e-1	2.7e-2	3.7e-1	3.3e-2	1.7e-3	1.5e-3	2.0e-3	4.4e-3	2.9e-2	1.4e-3	3.8e-4	1.5e-5
	$\langle \hat{\mathcal{C}}_k^i \rangle$	2.0e-1	3.3e-1	2.9e-2	3.6e-1	3.6e-2	2.0e-3	1.9e-3	2.3e-3	4.7e-3	3.0e-2	1.5e-3	4.0e-4	1.5e-5
$\langle \mathcal{C}_k^i \rangle / \langle \hat{\mathcal{C}}_k^i \rangle$		1.0	1.0	0.9	1.0	0.9	0.9	0.8	0.9	0.9	1.0	0.9	0.9	1.0
YTHAN	$\langle \mathcal{C}_k^i \rangle$	4.1e-1	2.3e-1	3.3e-2	2.2e-1	5.1e-2	3.0e-3	2.7e-3	2.8e-3	2.0e-3	3.6e-2	5.3e-3	1.1e-3	5.8e-5
	$\langle \hat{\mathcal{C}}_k^i \rangle$	3.7e-1	2.4e-1	3.9e-2	2.2e-1	5.6e-2	3.5e-3	4.8e-3	5.0e-3	3.0e-3	5.2e-2	8.1e-3	2.7e-3	7.5e-4
$\langle \mathcal{C}_k^i \rangle / \langle \hat{\mathcal{C}}_k^i \rangle$		1.1	1.0	0.9	1.0	0.9	0.8	0.6	0.6	0.6	0.7	0.6	0.4	0.1

Conclusion

Summary

We have extended the tractability of detecting network motifs. This allows for faster detection of larger motifs than previously possible.

To Do

There remains much room for more theory: Extending the direct calculation, building motifs from motifs, analyzing the algorithms.

What We Do

Implementing the algorithms, adding new functionalities and a graphical interface.

http://www.minet.uni-jena.de/~wernicke/motifs/