

## Efficient sampling algorithm for estimating subgraph concentrations and detecting network motifs

N. Kashtan<sup>1,3</sup>, S. Itzkovitz<sup>1,2</sup>, R. Milo<sup>1,2</sup>, U. Alon<sup>1,2</sup>

<sup>1</sup>*Department of Molecular Cell biology*, <sup>2</sup>*Department of Physics of Complex Systems*,  
<sup>3</sup>*Department of Computer Science & Applied Mathematics*,  
*Weizmann Institute of Science, Rehovot 76100, Israel*

### Abstract

Biological and engineered networks have recently been shown to display network motifs: a small set of characteristic patterns which occur much more frequently than in randomized networks with the same degree sequence. Network motifs were demonstrated to play key information processing roles in biological regulation networks. Existing algorithms for detecting network motifs act by exhaustively enumerating all subgraphs with a given number of nodes in the network. The runtime of such full enumeration algorithms increases strongly with network size. Here we present a novel algorithm that allows estimation of subgraph concentrations and detection of network motifs at a run time that is asymptotically independent of the network size. This algorithm is based on random sampling of subgraphs. Network motifs are detected with a surprisingly small number of samples in a wide variety of networks. Our method can be applied to estimate the concentrations of larger subgraphs in larger networks than was previously possible with full enumeration algorithms. We present results for high-order motifs in several biological networks and discuss their possible functions.

**Availability:** A software tool for estimating subgraph concentrations and detecting network motifs (*mfinder* 2.0) and further information is available at:

<http://www.weizmann.ac.il/mcb/UriAlon/>

**Contact:** uri.alon@weizmann.ac.il

### Introduction

Electronic networks are usually built out of recurring circuit elements such as operational amplifiers and filters. Recently, it was found that biochemical and neuronal networks share a similar property: they contain subgraphs that occur in the network far more often than in randomized networks (Milo 2002). Other types of networks such as ecological and technological networks contain different sets of characteristic network motifs. In the case of biological regulation networks, the network motifs were suggested to play key information processing roles (Shen-Orr 2002). In the transcription network of bacteria and yeast, for example, three major network motifs were found (Milo 2002; Shen-Orr 2002). One of these, the feed-forward loop, has been shown theoretically to perform information-processing tasks such as sign-sensitive filtering, response acceleration and pulse-generation (Mangan and Alon 2003). The sign-sensitive filtering function of the feedforward loop was then demonstrated experimentally using high-resolution gene expression measurements on the *ara* system of *E. coli* (Mangan 2003). A second network motif in transcription networks, the single-input module, has been shown theoretically (Shen-Orr 2002) and experimentally (Kalir 2001; Ronen 2002; Zaslaver 2003) to generate temporal programs of expression. In several systems, it was found experimentally that the temporal order of expression corresponds to the functional

order of the gene products in the system (Laub 2000; Kalir 2001; Ronen 2002; Zaslaver 2003). More generally, finding network motifs in a biological network raises the hope that the network function can be understood in terms of basic computational building blocks.

In order to detect network motifs, one needs to count the number of appearances of all types of  $n$ -node subgraphs in the network, as well as in an ensemble of randomized networks. There are many isomorphic types of subgraphs with a given number of nodes (there are 13 different types of connected, directed three node subgraphs, 199 four-node subgraphs, 9364 5-node subgraphs etc.). Motifs are those subgraphs which occur statistically significantly more often in the real network than in the randomized network. As a stringent control, the random network ensemble used preserves the single node characteristics of the real network: the number of incoming, outgoing and mutual edges for each node is preserved.

There are therefore two main tasks in detecting network motifs: 1. Generating an ensemble of proper random networks (Milo 2003) 2. Counting the subgraphs in the real network and in all random networks. Here we focus on the latter task.

Counting subgraphs in a large network is known to be a difficult computational task. Efficient algorithms are known for exact counting of certain classes of subgraphs such as cycles (Johnson 1975; Alon 1997) and cliques (Akkoyunlu 1973; Nesetril and Poljak 1985) reviewed in (Bezem and J. 1987). Approaches for approximate counting were developed in order to cope with the complexity of exact counting in other types of problems (Lovasz 1993; Jerrum and Sinclair 1996; Jerrum 2003). Several sampling algorithms were developed for enumeration of classical graph problems such as counting Hamiltonian cycles or spanning trees in graphs (Dyer 1994; Frieze and Kannan 1999; Jerrum 2003). Algorithms have been developed for finding frequent subgraphs that recur many times in a set of networks (Inokuchi 2000; Kuramochi and Karypis 2001). An approach for approximating frequencies of subgraphs in a given non-directed, labeled graph was developed by (Duke 1995), based on the regularity lemma of graphs (Szemerédi 1978; Alon 1994). This algorithm has strong constraints on the subgraph size for a given network size (on a typical biological network of hundreds to thousands of nodes this algorithm is bounded to 3-node subgraphs). The runtime of the algorithm grows polynomially with network size. Thus, there is a lack of practical algorithms, whose runtime does not scale with network size, for counting subgraphs in networks.

In previous work, we developed an exhaustive-search algorithm that counts all the subgraphs of a given number of nodes,  $n$ , in the network (Milo 2002; Shen-Orr 2002). For example, for  $n=3$ , the algorithm outputs the numbers of all 13 types of 3-node connected directed subgraphs. The performance of this algorithm for counting  $n$ -node subgraphs scales with the total number of  $n$ -node subgraphs in the network. The runtime, therefore, scales at least as the network size. The runtime is made even longer by the presence of hubs (highly connected nodes). Hubs combinatorially generate many subgraphs. The existence of hubs is a common feature of many natural and technological networks (Barabasi and Albert 1999). The number of subgraphs, and the algorithm runtime, also increase dramatically when we consider subgraphs with  $n \geq 5$ . Therefore, efficient algorithms are needed in order to count subgraphs in large networks.

In order to cope with the complexity of subgraph counting in large directed networks we present a probabilistic algorithm termed the ”*sampling method for subgraph counting*”. This algorithm does not exhaustively enumerate subgraphs, but instead samples subgraphs in order to estimate their relative frequency. The runtime of the algorithm asymptotically does not depend on the network size. Surprisingly few samples are needed to reliably detect network motifs. The *sampling method* is useful for analyzing very large networks or for detection of high-order motifs, which are beyond the reach of the previous exhaustive search algorithms.

## **METHODS**

### **Subgraph concentrations**

For simplicity in this study we will consider directed networks with one color of edges and nodes. The number of appearances of the subgraph type  $i$  is  $N_i$ . The concentration of  $n$ -node subgraph type  $i$  is the ratio between its number of appearances and the total number of  $n$ -node connected subgraphs in the network:

$$C_i = \frac{N_i}{\sum_i N_i}$$

For example the feed forward loop (FFL - subgraph M4 in table 2) appears 42 times in the *E. coli* gene transcriptional regulation network studied in (Shen-Orr 2002). The total number of 3-node connected subgraphs in the network is 5206 therefore the FFL concentration is  $C_{\text{FFL}}=42/5206=0.008$ .

### **Subgraphs sampling**

A specific subgraph is defined by a set of nodes in the graph. The edges of the subgraph are the edges between these nodes in the original graph. The algorithm samples subgraphs by picking random edges until a set of  $n$  nodes is reached. The following describes the random sampling procedure of one  $n$ -node subgraph from the network: Pick a random edge from the network, and then iteratively expand the subgraph (the nodes that are connected by the picked edges) by picking random neighboring edges until the subgraph reaches  $n$  nodes. For each random choice of an edge, in order to pick an edge that will expand the subgraph size by one, prepare a list of all such candidate edges and then choose randomly an edge from the list. Finally, the sampled subgraph is defined by the set of  $n$  nodes and of all the edges that connect between these nodes in the original graph (not just the edges which were picked by the expansion process). (See algorithm formal description in Fig 1).

Definitions:  $E_S$  is the set of picked edges.

$V_S$  is the set of all nodes that are touched by the edges in  $E$ .

Init  $V_S$  and  $E_S$  to be empty sets.

1. Pick a random edge  $e_1 = (v_i, v_j)$ . Update  $E_S = \{e_1\}, V_S = \{v_i, v_j\}$

2. Make a list  $L$  of all neighbor edges of  $E_S$ .

Omit from  $L$  all edges between members of  $V_S$ .

3. Pick a random edge  $e = (v_k, v_l)$  from  $L$ .

Update  $E_S = E_S \cup \{e\}, V_S = V_S \cup \{v_k, v_l\}$

4. Repeat steps 2-3 until completing an n-node subgraph (until  $|V_S| = n$ ).

5. Calculate the probability to sample the picked n-node subgraph.

**Fig 1: Sampling Algorithm.** Steps 1-5 represent a single sample; this is repeated  $S_T$  times.

### Exact correction for non-uniform sampling

A specific subgraph is a set of  $n$  connected nodes in the network. The probability to sample different specific subgraphs in the network is not equal. In order to correct for this we calculate the probability  $P$  of sampling the specific subgraph. Each subgraph type receives a score (the initial score of all subgraphs is set to zero). After each sample we add a weighted score of  $W=1/P$  to the score of the relevant subgraph type. This is repeated for a total number of samples  $S_T$ . Finally we calculate the concentrations of all subgraph types according to their scores.

In each sample of an  $n$ -node subgraph, an ordered set of  $n-1$  edges is iteratively randomly picked. In order to compute the probability  $P$  of sampling the subgraph we need to check all possible such ordered sets of  $n-1$  edges (denoted as  $(n-1)$ -permutations) which could lead to sampling of the subgraph. The probability of sampling the subgraph is the sum of the probabilities of all possible such ordered sets of  $n-1$  edges.

$\{v_1..v_n\}$ : subgraph nodes;  $\{e_1..e_m\}$ : subgraph edges; where  $m \geq n-1$

$S_m$ : Set of all  $(n-1)$ -permutations of the edges from the set  $\{e_1..e_m\}$  that could lead to a sample of the subgraph.

$E_j$  is the  $j$ -th edge in a specific  $(n-1)$ -permutation.

$$P = \Pr[\text{subgraph } \{v_1..v_n\} \text{ picked}] = \sum_{\sigma \in S_m} \prod_{E_j \in \sigma} \Pr[E_j = e_j | (E_1..E_{j-1}) = (e_1..e_{j-1})]$$

The conditional probability of picking the next edge at each iteration in an  $(n-1)$ -permutation of edges depends on the size of the candidate edges lists (the number of neighboring edges that can expand the size of the subgraph by one).

Each sample is given the following weight:

$$W(\{v_1..v_n\}) = \frac{1}{\Pr[\text{subgraph } \{v_1..v_n\} \text{ picked}]}$$

In figure 2 we illustrate this procedure on a simple example network. The two specific subgraphs considered in this example, nodes  $\{1,2,3\}$  and nodes  $\{4,5,6\}$ , have different sampling probabilities and are assigned different weights in order to ensure uniform counting.

### Calculating the concentrations of n-node subgraphs

For every sample we add the weighted score  $W$  to the accumulated score  $S_i$  of the relevant subgraph type  $X_i$ :  $S_i = S_i + W$ . After  $S_T$  samples we calculate the subgraphs concentrations ( $C_i$ ). Assuming we sampled  $L$  different subgraph types, then:

$$C_i = S_i / \sum_{k=1}^L S_k$$

All runtime analysis was done on a 1.7GHz Pentium 4 CPU with 1GB RAM. The loading time of the network was not included.

## Results

### Comparing *sampling method* results with full enumeration

In table 1 we show the results of the *sampling method* with different number of samples for 3-node subgraphs on a  $3.25 \times 10^5$  node,  $1.46 \times 10^6$  edge WWW network (Barabasi and Albert 1999). Total number of 3-node subgraphs in the network is  $2.87 \times 10^6$ . Running the algorithm with as few as 5000 samples gives a good approximation for all thirteen 3-node subgraph concentrations (Table 1). Even with 5000 samples, the 5 different network motifs (shaded subgraphs) are detected as significant vs. randomized networks, due to their high Z-scores ( $Z = (C_{real} - \langle C_{rand} \rangle) / \sigma_{rand}$  where  $C_{real}$  is the concentration in the real network,  $\langle C_{rand} \rangle$  and  $\sigma_{rand}$  are the mean and SD in the randomized networks) (Milo 2002). The runtime was about 500 fold faster than the full enumeration algorithm.

In table 2 we show the results of the *sampling method* for subgraphs with  $n=3,4,5$  in a biological regulatory network. We present the results for all the 3-nodes subgraphs that appear in the transcriptional regulation network of *E. coli* (Shen-Orr 2002), as well as the 4 and 5-node subgraphs which are network motifs. It can be seen that the *sampling method* estimates the subgraphs concentration very accurately even for subgraphs with relatively low concentration (for example, 5-node motifs with  $c=10^{-5}$ ).

Generally, we find that in a variety of networks, network motifs were found to have relatively high concentrations. Most motifs of size 3-4 have  $C_i > 10^{-3}$ , 5 nodes motifs usually have  $C_i > 10^{-5}$ . This means that the sampling algorithm should prove especially effective for motif detection.

### Runtime complexity analysis

The main cost in steps 1-4 of the *sampling method* (Fig 1) is to maintain the list of edges from which the next random edge should be picked in each step of the sampling. In the worst case the list length is dominated by the hub degree ( $D$ ), where  $D$  is the maximal number of edges per node in the network. Maintaining the list includes omitting recurring edges and throwing away edges between the current set of nodes that were already picked. Worst complexity is  $O(Dn)$  for every sample of  $n$ -node subgraph. By maintaining an efficient data structure this complexity can be reduced to  $O(n^2)$  (see Appendix 1).

We now estimate the complexity of calculating the probability of sampling a specific  $n$ -node subgraph (Fig 1 - step 5): In a single  $(n-1)$ -permutation of the subgraph edges  $\{e_1..e_m\}$ , for every edge we need to calculate the probability to sample the next edge. In order to do this we need to calculate the effective degree of each node at each step

of picking the next random edge. By the degree of each node this can be done in  $O(n)$  operations. Because there are  $(n-1)$  steps of such iterations we get  $O(n^2)$ . In sparse networks, the number of edges,  $m$ , in a connected subgraph is typically  $n-1 \leq m \leq Kn$  ( $K$  is a small constant,  $K < n$ ). Thus the number of  $(n-1)$ -permutations of edges is of the order of:  $O(n^{n-1})$ . In total we get a complexity per sample of  $O(n^2) \times O(n^{n-1}) = O(n^{n+1})$ . We conclude that the total run time of the algorithm is  $R_S = S_T \times [O(n^{n+1})]$ . This qualitatively agrees with runtime measurements for sampling subgraphs of sizes 3-8 (Fig 3) on the transcriptional regulation network of *E. coli*.

### Analyzing runtime of the *sampling method* vs. exhaustive search

We would like to evaluate the ratio  $r$  of the runtime of the exhaustive search algorithm ( $R_E$ ) and the runtime of the *sampling method* ( $R_S$ ). The runtime of exhaustive search algorithms is mainly dominated by the total subgraphs number; therefore its complexity is  $\Omega(n^2 T)$  where  $T$  is the total number of  $n$ -node subgraphs ( $n^2$  is the minimal complexity of analyzing the adjacency matrix of a subgraph of size  $n$ ). The total number of  $n$ -node subgraphs in networks that contain a hub is dominated by the hub degree ( $D$ ) and can be roughly evaluated by  $T = D^{n-1}$  (Itzkovitz 2003). For such networks we get a runtime of  $\Omega(n^2 D^{n-1})$ . Runtime dependence on network size ( $N$ ) comes from its effect on  $D$ . In networks without hubs the total number of subgraphs can be roughly evaluated by  $T = N < d >^{n-1}$  where  $< d >$  is the average degree of the nodes.

For networks that contain a hub the runtime ratio is

$$r = \frac{R_E}{R_S} = \Omega\left(\frac{n^2 D^{n-1}}{n^{n+1}} \cdot \frac{1}{S_T}\right) = \Omega(D/n)^{n-1} \cdot \frac{1}{S_T}$$

For a network with hub degree  $D=1000$ , and  $S_T=10^5$  samples, we find for 3-node subgraphs  $r \sim 1$ , for 4-node subgraphs  $r \sim 150$  and for 5-node subgraphs  $r \sim 1.5 \times 10^4$ . We find that for subgraphs of 4 nodes and above the runtime of the *sampling method* is much smaller than an exhaustive search algorithm (Fig 4).

For a network that does not have hubs the ratio is

$$r = \frac{R_E}{R_S} = \Omega\left(\frac{n^2 N < d >^{n-1}}{n^{n+1}} \cdot \frac{1}{S_T}\right) = \Omega(< d > / n)^{n-1} \cdot \frac{N}{S_T}$$

For such a network with  $N=10,000$ ,  $< d >=3$  and  $S_T=10^5$ , we find for 3-node subgraph  $r \sim 0.1$ , for 4-node subgraph  $r \sim 0.05$  and for 5-node  $r \sim 0.01$ .

We conclude that for networks without hubs, the runtime of the *sampling method* is not smaller than the exhaustive search algorithm. However it can be useful in this case to run the *sampling method* with a small number of samples to get a low accuracy subgraph concentration profile of large non-hubs networks, or for the purpose of detection of strong network motifs.

In order to compare the runtime of the two algorithms, we generated synthetic scale free networks (exponent  $\gamma = 2.01$ ) with varying number of nodes  $N$ , using the methods of (Itzkovitz 2003) (Fig 4). All the networks have the same average connectivity ( $< d >=2.4$ ). We set the hub degree to be  $D=0.1N$ . The runtime of the exhaustive search algorithm scales as the total number of subgraphs. Since the total number of  $n$ -node subgraphs scales as  $D^{n-1}$ , and  $D$  scales with  $N$ , the runtime of the direct enumeration method increases polynomially with the network size as  $N^{n-1}$ . The

runtime of the *sampling method*, in contrast, is almost independent of the network size or hub degree (for a constant number of samples). The relative advantage of the *sampling method* becomes more significant as network size increases.

### **Algorithm Convergence**

The problem of deciding “how many samples are enough?” was well explored in random sampling from databases (Flajolet and Martin 1985; Olken and Rotem 1995; Chaudhuri 1998; Gibbons 2001) and estimating statistics on a sampled population (Bunge and Fitzpatrick 1993). It was shown to be a hard problem (Chaudhuri 1998). The number of samples required for good estimation with high probability is hard to approximate when the concentration distribution is not known a-priori. In simulations, we analyzed the results of the *sampling method* as a function of the number of samples (Fig 5A-D). The subgraph concentrations calculated by the sampling algorithm converge to the fully enumerated concentrations. Different numbers of samples were required for achieving good estimations for different subgraphs and in different networks. Nevertheless all the simulations we performed, on a variety of networks, showed that the results converge towards the real values within  $S_T=10^5$  samples or less (Fig 5A-D). It is seen that even with a small number of samples one can reliably estimate concentration as low as  $C=10^{-5}$ . It is possible to use convergence studies in order to decide the required number of samples, as described in Appendix 2.

### **Discussion**

#### ***Sampling method* allows accurate counting of rare, high-order subgraphs and motifs**

We have presented a sampling algorithm to estimate subgraph concentrations in a network. The sampling algorithm employs analytical corrections for sampling biases. The runtime of this algorithm is asymptotically independent of network size. The algorithm is thus far more efficient, for the commonly occurring networks with hubs, than full-enumeration approaches.

The *sampling method* is able to detect subgraphs whose concentration is very low with relatively few samples (for example, the concentration of motifs with  $c=10^{-5}$  can be accurately estimated with only 50,000 samples, table2 - subgraphs M9,M10). This effect is due to the presence of hubs in the networks. We can divide specific subgraphs in the network into two types, according to their probability to be sampled by the algorithm. The first type, which we refer to as ‘non-hub subgraphs’, are all subgraphs that either do not contain a hub node or contain a single hub node but the other  $n-1$  nodes remain connected if the hub and its edges are removed. The second type, which we refer as ‘hub subgraphs’, are all other subgraphs in the network. ‘Hub subgraphs’, which are typically dominated by many hub edges (i.e. edges touching a hub) are characterized by a small probability to be sampled. The reason for the small probability is that for every  $(n-1)$ -permutation we necessarily reach the hub before we complete an  $n$ -node set. Therefore for all possible  $(n-1)$ -permutations the candidate edges list is large (on the order of the hub degree) at least in one of the iterations, which leads to a small sampling probability. This effect becomes stronger with larger subgraph size. In contrast, ‘non-hub subgraphs’ have a higher probability to be sampled because there exists at least one option to sample the subgraph without reaching the hub or with reaching it last (when the hub is the  $n$ -th node to be reached). These ‘non-hub subgraphs’ can be picked up with even a relatively small number of

samples, and are given a small weight by the analytical sampling bias correction made by the algorithm. We conclude that: a) The probability to sample ‘non-hub subgraphs’ is higher than ‘hub subgraphs’, and therefore such subgraphs (although they may be rare) can be sampled with a much smaller number of samples than expected based on their concentration. b). ‘Hub subgraphs’ have a lower probability to be sampled, but this is usually compensated by their high relative concentration. In both cases the correction for the non-uniformity assures that the concentration estimation is correct. Fast convergence rate is achieved in both cases due to the higher probability of sampling ‘non-hub subgraphs’ and due to the high concentration of ‘hub subgraphs’.

In particular, network motifs are reliably detected by the algorithm with a surprisingly small number of samples. This reflects the fact that in the networks we have analyzed (Milo 2002), the motifs are distributed throughout the network and not only near hubs. This sampling advantage of the method contributes to the efficiency of the algorithm in estimating subgraphs concentration and in network motif detection.

### **Even a relatively small number of samples can be enough to detect motifs**

Applying the method with a small number of samples can still be sufficient to detect network motifs. In order to detect motifs we look for subgraphs that appear in the network significantly more than in randomized networks. We use the sampling method to estimate subgraphs concentrations both in the real network and in the randomized networks. Most of the motifs we found, especially in large networks, tend to have high Z-scores ( $Z = (C_{real} - \langle C_{rand} \rangle) / \sigma_{rand}$ ) compared to random networks subgraphs statistics. The Z-scores of network motifs tend to be higher the larger the subgraph size and the larger the network. Thus, for large networks and subgraphs, a high cutoff of  $Z=5$  or  $10$  can be used to detect significance using the sampling algorithm. Setting the Z-score cutoff to high values is important also for avoiding false positives (this can happen when  $\langle C_{rand} \rangle$  is underestimated due to very low concentrations in the randomized networks, relatively to the number of samples) while not missing interesting motifs. The observed high Z-scores of network motifs usually assures that with a small number of samples the difference between the appearance of motifs in the real and random networks is large enough to be detected, even when the *sampling method* cannot provide very high accuracy for the actual concentrations. In typical cases, sampling sufficient to provide two-fold errors should be enough for purposes of network motif detection.

### **Motif generalizations in the E. coli transcription network:**

We employed the algorithm to detect large motifs in networks where we have previously analyzed only  $n=3$  and  $4$ . In the E. coli network, the only 3-node motif is the feedforward loop (Table 2, M4). The feedforward loop was suggested to have a specific biological function in transcription networks. Feedforward loops with positive regulations have been shown experimentally to function as a sign sensitive delay element (Shen-Orr 2002; Mangan 2003). With other sign combinations, the feedforward loop can function as a pulse-generator or response accelerator (Mangan and Alon 2003). At the level of 4-node subgraphs, a motif appears which is a feedforward loop with 2 output nodes (M6). At the level of 5-node subgraphs, a feedforward loop with 3 outputs appears (M7). This suggests the proper generalization of the feedforward loop is a motif with  $n$ -output nodes (Kashtan 2003) (table 3 L1,L6). Similarly, the 4-node bi-fan motif (M5) generalizes at the level of 5-



node motifs to patterns with two inputs and three outputs (M8), or three inputs and two outputs (M9). These generalize at higher order subgraphs (table 3 L2,L3,L7) to the motif termed ‘dense overlapping regulons’ (Shen-Orr 2002). These structures are hard-wired combinatorial decision-making circuits. Additional high order motifs are summarized in table 3. It can be seen that most motifs are constructed by smaller motifs following generalization rules (table 3 L1,L6,L2,L3,L7) or by combining few motifs together (table 3 L4,L5,L8-10). This suggests that small motifs and their generalizations can be thought of as the basic building blocks of this network.

### **High order motifs in the neuronal network of the worm *C. elegans***

This network describes synaptic connections between neurons in *C. elegans*. Two neurons are connected if at least one synaptic connection exists between them. Applying the full enumeration algorithm we have previously detected 3 and 4 nodes motifs (Milo 2002). We applied the *sampling method* on the neuronal network of *C. elegans* for 5 and 6 nodes motifs detection, which was beyond reach using the full enumeration algorithm. We find in the neuronal network of *C. elegans* a different generalization form of the feedforward loop – the multi-input feedforward loop (table 4, E1). This generalization is suggested to act as an integration unit of several inputs (sensory neurons) and can preserve the basic function of the feed forward loop as persistence detector (Kashtan 2003). A multi output generalization of the ‘regulating mutual’ (a 3-node motif in this network) appears (table 4 E3, E10). We find other significant structures which are formed from combinations of 3 and 4-node motifs (E2, E4). In addition we find generalized forms of the *bi-parallel* (a 4-node motif found in this network): these are the double-input (E5) and double-output (E7) *bi-parallel* generalizations. These two motifs are similar in structure to two-layer perceptrons (feed-forward neural network). Two-layer perceptrons have been shown to be able to implement complex functions such as XOR (Exclusive OR) which cannot be implemented by single layer perceptrons (Hertz 1991). Multi-layers circuits (E5-E8, E11-E13) can perform complex computations using different weights on the edges and different input functions on the nodes.

It would be interesting to apply network motif analysis, assisted by tools such as the *sampling method*, to metabolic (Ouzounis and Karp 2000; Wagner and Fell 2001), signaling, immunological and other biological networks.

The ability to estimate the subgraph content of a network may be useful in a number of fields. For example, solving short-time diffusion or transport problems on networks (Lovasz 1993; Bosiljka and Rodgers 2002; Kim 2003) will be aided by knowledge of the local structure statistics. For motif detection, these algorithms enable the analysis of much larger networks and larger subgraphs than was previously feasible.














**Acknowledgments:** We thank N. Alon, S. Holmes, M. Naor, M.E.J. Newman, R. Raz, R. Shamir and all members of our lab for discussions. We acknowledge support by the James and Ilene Natan fund, the Harry M. Ringel Memorial Fund and the Israel Science Foundation.

## References







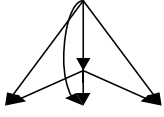
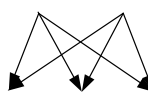
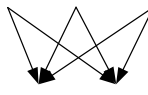
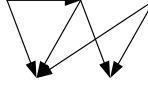
- Achacoso, T. B. and Yamamoto, W. S. (1992). *AY's Neuroanatomy of C. elegans for Computation*, CRC Press.
- Akkoyunlu, E. (1973). The enumeration of maximal cliques of large graphs. *SIAM Journal on Computing* **2**(1): 1-6.
- Alon, N., Duke, R., Lefmann, H., Rödl, V. and Yuster, R. (1994). The Algorithmic Aspects of the Regularity Lemma. *J. of Algorithms* **16**(1): 80-109.
- Alon, N., Yuster, R. and Zwick, U. (1997). Finding and counting given length cycles. *Algorithmica* **17**: 209-223.
- Barabasi, A.-L. and Albert, R. (1999). Emergence of Scaling in Random Networks. *Science* **286**: 509-512.
- Bezem, G. J. and Van Leeuwen, J. (1987). Enumeration in graphs. *Utrecht, Universiteit Utrecht*.
- Bosiljka, T. and Rodgers, G. J. (2002). Packet Transport on Scale Free Networks. *Advances in Complex Systems* **5**: 445-456.
- Bunge, J. and Fitzpatrick, M. (1993). Estimating the number of species: A review. *J. of the American Statistical Association* **88**: 364-373.
- Chaudhuri, S., Motwani, R. and Narassaya, V. (1998). Using Random sampling for Histogram Construction: How much is enough? *ACM SIGMOD Conf.* 436-447.
- Costanzo, M. C., Crawford, M. E., Hirschman, J. E., Kranz, J. E., Olsen, P., Robertson, L. S., Skrzypek, M. S., Braun, B. R., Hopkins, K. L., Kondu, P., Lengieza, C., Lew-Smith, J. E., Tillberg, M. and Garrels, J. I. (2001). YPD, PombePD and WormPD: model organism volumes of the BioKnowledge library, an integrated resource for protein information. *Nucleic Acids Res* **29**(1): 75-9.
- Duke, R., Lefmann, H. and Rödl, V. (1995). A fast approximation algorithm for computing the frequencies of subgraphs in a given graph. *SIAM Journal on Computing* **24**(3): 598-620.
- Dyer, M., Frieze, A. M. and Jerrum, M. (1994). Approximately Counting Hamilton Cycles in Dense Graphs. *SODA*: 336-343.
- Flajolet, P. and Martin, G. (1985). Probabilistic counting algorithms. *J. of Comp. and Sys. Sci* **31**: 182-209.
- Frieze, A. and Kannan, R. (1999). Quick Approximation to Matrices and Applications. *Combinatorica* **19**(2): 175-220.
- Gibbons, P. B. (2001). Distinct Sampling for Highly-Accurate Answers to Distinct Values Queries and Event Reports. *VLDB*: 541-550.
- Hertz, J., Krogh, A. and Palmer, R. G. (1991). Introduction to the theory of neural computation. *Addison-Wesley Publishing company*.
- Inokuchi, A., Washio, T. and Motoda, H. (2000). An Apriori-based Algorithm for Mining Frequent Substructures from Graph Data. *Proc. of PKDD2000: The Fourth European Conference on Principles and Practice of Knowledge Discovery in Databases*.
- Iitzkovitz, S., Milo, R., Kashtan, N., Ziv, G. and Alon, U. (2003). Subgraphs in Random Networks. *Phys Rev E* **68**.026127 :
- Jerrum, M. (2003). Counting, sampling and integrating: Algorithms and Complexity. *Lectures in Mathematics ETH Zurich*.

- Jerrum, M. and Sinclair, A. (1996). The Markov Chain Monte Carlo method: An approach to approximate counting and integration. *Approximation Algorithms for NP-hard Problems*: 482-520.
- Johnson, D. B. (1975). Finding all the elementary circuits of a directed graph. *SIAM J. on Computing* **4**: 77-84.
- Kalir, S., McClure, J., Pabbaraju, K., Southward, C., Ronen, M., Leibler, S., Surette, M. G. and Alon, U. (2001). Ordering genes in a flagella pathway by analysis of expression kinetics from living bacteria. *Science* **292**(5524): 2080-3.
- Kashtan, N., Itzkovitz, S., Milo, R. and Alon, U. (2003). Network motifs in biological networks: Roles and Generalizations. *Under review*.
- Kim, B. J., Hong, H. and Choy, M. Y. (2003). Quantum and classical diffusion in small-world networks. *cond-mat/0306234 (To appear in PRB)*.
- Kuramochi, M. and Karypis, G. (2001). Frequent subgraph discovery. *In Proc. of the 1st IEEE International Conference on Data Mining (ICDM)*.
- Laub, M. T., McAdams, H. H., Feldblyum, T., Fraser, C. M. and Shapiro, L. (2000). Global analysis of the genetic network controlling a bacterial cell cycle. *Science* **290**(5499): 2144-8.
- Lovasz, L. (1993). Random walks on graphs: A survey. *Combinatorics Paul Erdos is eighty (Volume 2)*: 1-46.
- Mangan, S. and Alon, U. (2003). Structure and function of the feed-forward loop network motif. *Proc Natl Acad Sci U S A* **100**(21): 11980-5.
- Mangan, S., Zaslaver, A. and Alon, U. (2003). The Coherent Feedforward Loop Serves as a Sign-sensitive Delay Element in Transcription Networks. *J Mol Biol* **334**(2): 197-204.
- Milo, R., Kashtan, N., Itzkovitz, S., Neuman, M. E. J. and Alon, U. (2003). Uniform generation of random networks with arbitrary degree sequence. *cond-mat/0312028*.
- Milo, R., Shen-Orr, S., Itzkovitz, S., Kashtan, N., Chklovskii, D. and Alon, U. (2002). Network Motifs: Simple Building Blocks of Complex Networks. *Science* **298**: 824-827.
- Nesetril, J. and Poljak, S. (1985). On the complexity of the subgraph problem. *Commen. Math. Univ. Carol.* **26**: 415-419.
- Olken, F. and Rotem, D., . (1995) Random Sampling from Databases - A survey. *Statistics & Computing* **5**(1): 25-42.
- Ouzounis, C. and Karp, P. (2000). Global properties of the metabolic map of Escherichia coli. *Genome Res.* **10**(4): 568-76.
- Ronen, M., Rosenberg, R., Shraiman, B. I. and Alon, U. (2002). Assigning numbers to the arrows: parameterizing a gene regulation network by using accurate expression kinetics. *Proc Natl Acad Sci U S A* **99**(16): 10555-60.
- Shen-Orr, S., Milo, R., Mangan, S. and Alon, U. (2002). Network motifs in the transcriptional regulation network of Escherichia Coli. *Nature Genetics* **31**(1): 64-68.
- Szemerédi (1978). Regular partitions of graphs. *Proc. Colloque Inter. CNRS, J.-C. Bermond, J.-C. Fournier, M. Las Vegas, and D. Sotteau.*: 399-401.
- Wagner, A. and Fell, D. (2001). The small world inside large metabolic networks. *Proc R Soc Lond B Biol Sci.* **268**(1478): 1803-10.
- White, J., Southgate, E., Thomson, J. and Brenner, S. (1986). The structure of the nervous system of the nematode *Caenorhabditis elegans*. *Phil. Trans. Roy. Soc. London Ser. B* **314**: 1-340.

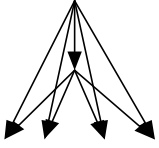
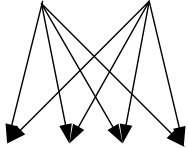
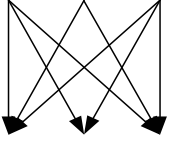
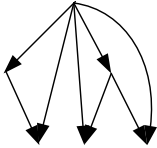
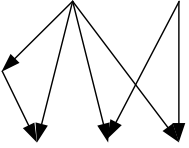
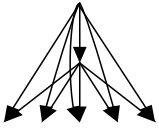
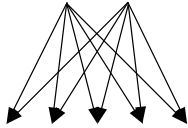
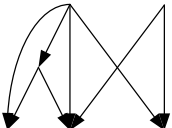
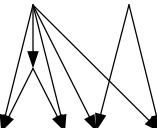
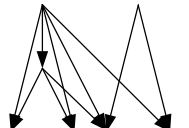
- Williams, R. J. and Martinez, N. D. (2000). Simple rules yield complex food webs. *Nature* **404**(6774): 180-3.
- Zaslaver, A., Mayo, A. and Alon, U. (2003). Just in time transcription programs in metabolic pathways. *Under review*.

Subgraph	Exhaustive Search		Sampling method		
	Total No. of Subgraphs <b>287M</b> <b>(Runtime: 2.9 hours)</b>		No. of samples <b>5K</b> <b>(Runtime: 15 sec)</b>	No. of samples <b>50K</b> <b>(Runtime: 37 sec)</b>	No. of samples <b>2.5M</b> <b>(Runtime: 28 min)</b>
ID	Appearances	Concentration (x 10 <sup>-3</sup> )	Concentration (x 10 <sup>-3</sup> )	Concentration (x 10 <sup>-3</sup> )	Concentration (x 10 <sup>-3</sup> )
6 	47015127	163.8	181.2	168.4	162.7
12 	2319911	8.1	10.3	6.7	8.2
14 	1363964	4.8	6.0	4.9	4.8
36 	218449147	761.0	732.2	754.8	762.2
38 	499763	1.74	1.97	1.75	1.73
46 	1164456	4.1	4.9	4.1	4.1
74 	4049373	14.1	17.4	15.7	13.9
78 	4954123	17.3	18.5	17.7	17.2
98 	9474	0.030	0.006	0.048	0.030
102 	40607	0.14	0.08	0.16	0.14
108 	309167	1.08	1.08	1.08	1.08
110 	106614	0.37	0.51	0.37	0.37
238 	6779926	23.6	25.9	24.2	23.5

**Table 1: Sampling method versus full enumeration on a WWW network.** Results of the *sampling method* of 3-node subgraphs compared to the full enumeration results, on a WWW network of the nd.edu domain (Barabasi and Albert 1999). The nodes represent web pages and the edges represent directed hyperlinks between pages. All 13 3-node connected subgraphs appear in the network. It can be seen that as few as 5000 samples (out of 287 million 3-node subgraphs) already give quite a good approximation of all the subgraph concentrations. Highlighted subgraphs were found to be network motifs.

Subgraph size	Subgraph		Full Enumeration		Sampling Method	
	ID	Shape	Appearances (Zscore)	Concentration (x 10 <sup>-3</sup> )	Concentration (x 10 <sup>-3</sup> ) (Zscore)	Num of Samples
<b>3</b>	S1		4777	917.60	916.60	1K  (~5K total 3-node subgraphs)
	S2		160	30.73	31.13	
	S3		227	43.60	43.64	
	M4		42 (z=10)	8.07	8.69 (z=10)	
<b>4</b>	M5		209 (z=9)	2.49	2.69 (z=8)	10K  (~85K total 4-node subgraphs)
	M6		51 (z=15)	0.61	0.65 (z=15)	
<b>5</b>	M7		54 (z=120)	0.038	0.035 (z=30)	50K  (~1.4M total 5-node subgraphs)
	M8		271 (z=16)	0.189	0.196 (z=11)	
	M9		20 (z=18)	0.014	0.013 (z=8)	
	M10		18 (z=12)	0.013	0.014 (z=8)	

**Table 2: Subgraphs of size 3-5 in the transcriptional regulation network of E. coli.**  
Results of the *sampling method* versus full enumeration for subgraphs size 3-5. The network is the transcriptional regulations network of E. coli (Shen-Orr 2002). For size n=4 and n=5 only motifs are shown. Statistical significant is represented by Z-score ( $Z = (C_{real} - \langle C_{rand} \rangle) / \sigma_{rand}$ ). It can be seen that the sampling method gives very accurate approximation with a relatively small number of samples. 5-node subgraphs although appearing in low concentrations show good results with 50K samples - while the total number of 5-nodes subgraphs is 1.4M. All the motifs detected by the full enumeration are detected by the *sampling method* (using  $Z > 5$ ).

Motifs of 6 and 7 nodes in E. coli transcriptional network									
L1	L2	L3	L4	L5					
									
C=0.002 Z=17	C= 0.015 Z=14	C= 0.005 Z=20	C= 0.006 Z=11	C= 0.077 Z=11					
L6	L7	L8	L9	L10					
									
C<0.000 Z=NA	C=0.001 Z=30	C= 0.008 Z=16	C= 0.003 Z=210	C= 0.002 Z=NA					

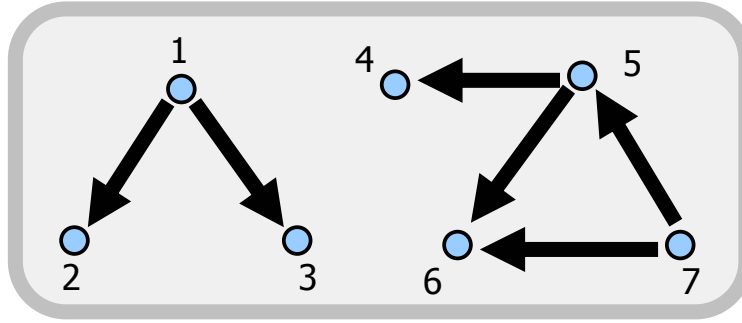
**Table 3: High order motifs (6 and 7 nodes) in E. coli transcription network.** The table summarizes the significant high order motifs in this network. We ran the sampling method with 200,000 and 500,000 samples for detecting 6-node and 7-node motifs respectively. Detection of 6-node and 7-node motifs in this network using the full enumeration algorithm was beyond reach. Concentrations ( $\times 10^{-3}$ ) ('C') and Z-scores ('Z') of the motifs are shown. 'NA': in the random networks not even one appearance of this subgraph was detected, therefore the Z-score could not be estimated.

Motifs of 5 and 6 nodes in <i>C. elegans</i> neuronal network											
5	E1	E2	E3	E4							
					C= 0.071	Z=12	C= 0.406	Z=21	C= 0.324	Z=230	C= 0.231
5	E5	E6	E7	E8							
					C= 0.170	Z=20	C= 0.420	Z=180	C= 0.343	Z=22	C= 0.687
6	E9	E10	E11	E12	E13	E14					
							C=0.018 Z=NA	C=0.059 Z=NA	C=0.166 Z=110	C=0.049 Z=NA	C=0.093 Z=100

**Table 4: High order motifs (5 and 6 nodes) in the neuronal network of the worm *C. elegans*.** Nodes represent neurons and edges represent synaptic connectivity. These motifs were detected by the sampling algorithm with 100,000 samples (on the real and random networks). Detection of 5-node and 6-node motifs in this network using the full enumeration algorithm was beyond reach. Concentrations ( $\times 10^{-3}$ ) ('C') and Z-scores ('Z') of the motifs are shown. 'NA': in the random networks not even one appearance of this subgraph was detected, therefore the Z-score could not be estimated. We note that the presented motifs are only a partial list of all the 5-node and 6-node motifs that were detected.

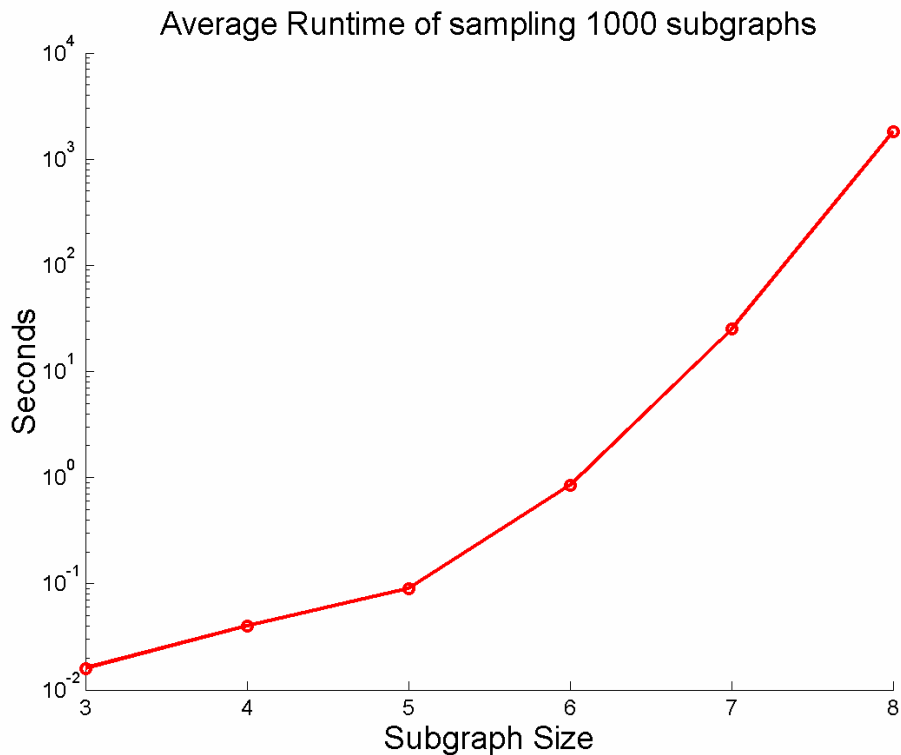


**Example Network:**

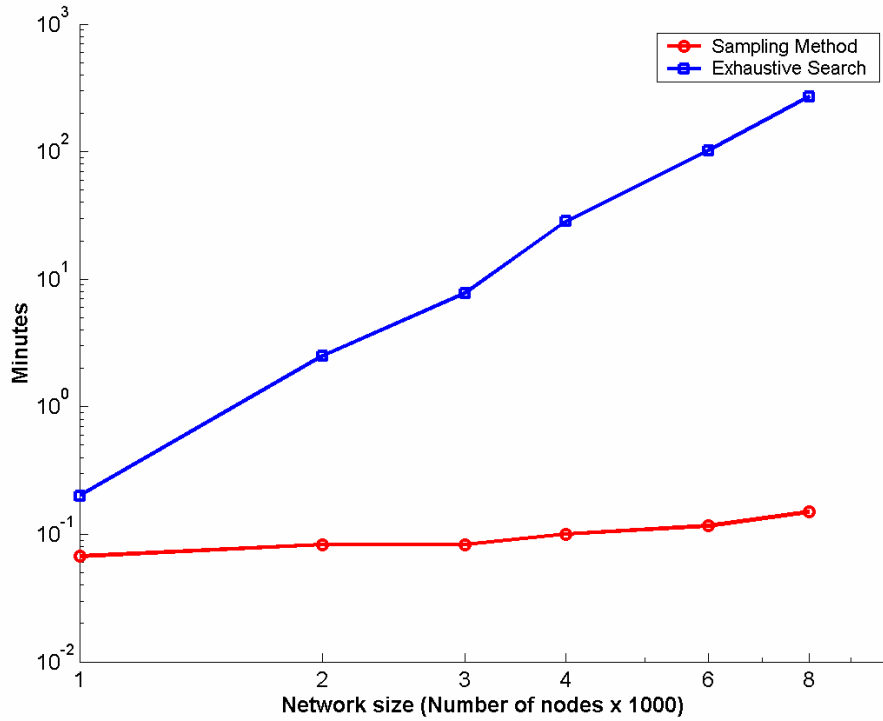


<p><b>Probability to sample {1,2,3}:</b>          There are 2 possibilities to sample {1,2,3}:          1. Pick first (1,2): <math>Pr=1/E=1/6</math>.          then pick (1,3): <math>Pr=1</math>.  <math>Pr[(1,2) \text{ then } (1,3)] = 1/6 * 1 = 1/6</math>.          2. Pick first (1,3): <math>Pr=1/E=1/6</math>.          then pick (1,2): <math>Pr=1</math>.  <math>Pr[(1,3) \text{ then } (1,2)] = 1/6 * 1 = 1/6</math>.  <b>In Total: <math>Pr\{1,2,3\} = 1/6 + 1/6 = 1/3 = 12/36</math></b></p>	<p><b>Probability to sample {4,5,6}:</b>          There are 2 possibilities to sample {4,5,6}:          1. Pick first (5,4): <math>Pr=1/E=1/6</math>.          then pick (5,6): <math>Pr=1/2</math>.  <math>Pr[(5,4) \text{ then } (5,6)] = 1/6 * 1/2 = 1/12</math>          2. Pick first (5,6): <math>Pr=1/E=1/6</math>.          then pick (5,4): <math>Pr=1/3</math>.  <math>Pr[(5,6) \text{ then } (5,4)] = 1/6 * 1/3 = 1/18</math>.  <b>In Total: <math>Pr\{4,5,6\} = 1/12 + 1/18 = 5/36</math></b></p>
---	---

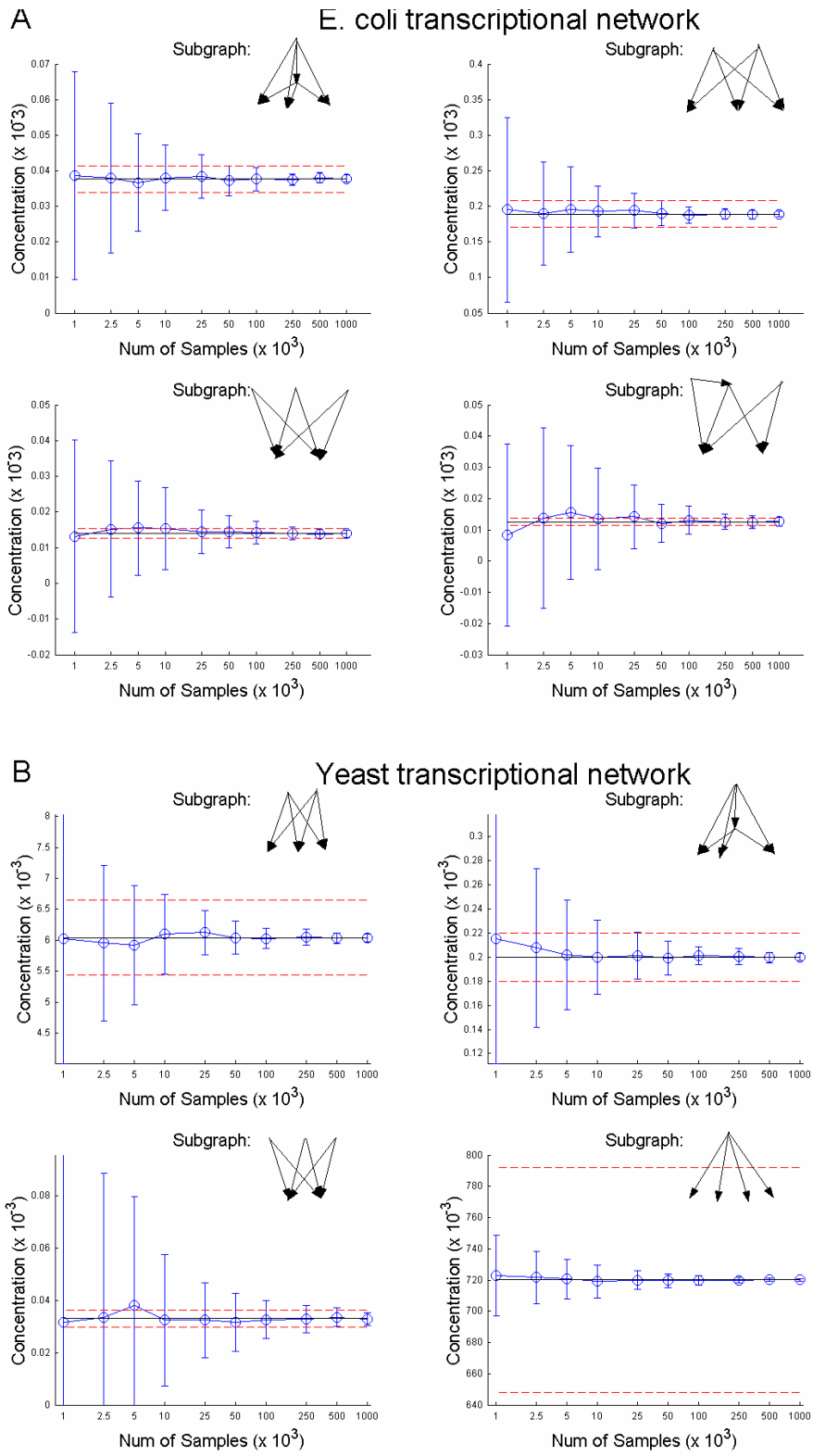
**Fig 2: Different probabilities of sampling different subgraphs.** Example of a toy network of 7 nodes and 6 directed edges. The probabilities to sample two different 3-nodes subgraphs are different although they both are of the same subgraph type (V-shaped outgoing edges).



**Fig 3: Runtime per 1000 samples for different subgraph sizes: 3-node up to 8-node subgraphs (semi-log scale).** The network analyzed is the transcriptional regulation of E. coli (Shen-Orr 2002). The scaling of the runtime of the *sampling method* follows the theoretical analysis of  $O(n^{n+1})$  where n is the subgraph size.

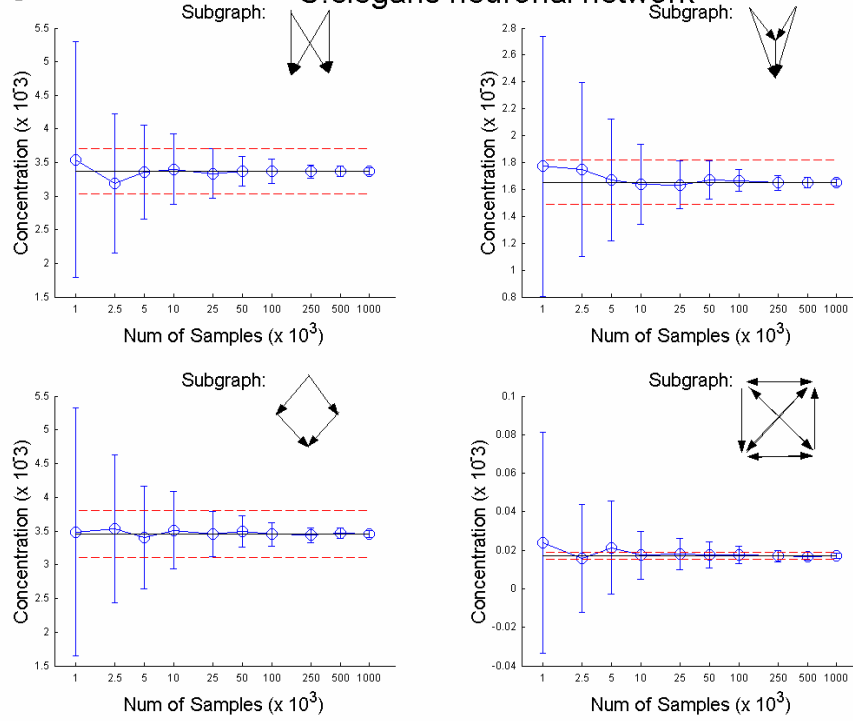


**Fig 4: Runtime of the *sampling method* vs. exhaustive search as a function of network size (log-log scale).** The networks are synthetic scale free networks ( $\gamma = 2.01$ ) with equal average connectivity ( $\langle d \rangle = 2.4$ ). The hub degree is 10% of total number of nodes. The *sampling method* was run with 100,000 samples for all the networks. The runtime of the exhaustive search scales as the total number of subgraphs while the runtime of the *sampling method* is almost constant.



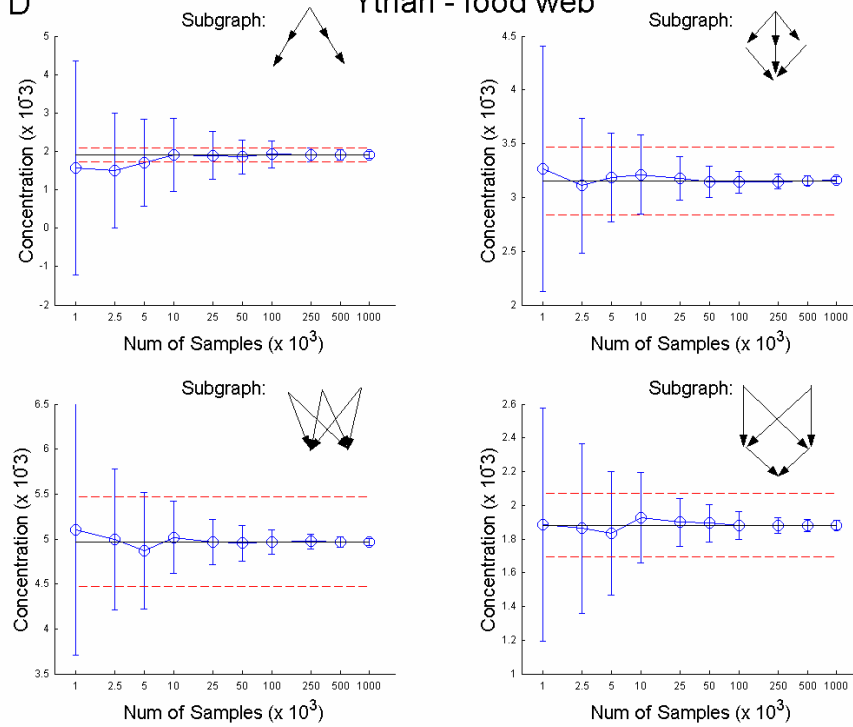
C

C. elegans neuronal network



D

Ythan - food web



**Fig 5: Convergence of the *sampling method* results on different networks.** Concentrations calculated by the *sampling method* for different subgraphs on different networks as a function of number of samples. The true concentration was found by full enumeration (black line). We compare the results of the *sampling method* with different number of samples. We ran the algorithm 100 times for each no. of samples ( $S_T$ ) on each one of the networks. The average concentration and standard deviation are shown (Blue). Real concentrations  $\pm 10\%$  are shown in dashed red. It can be seen that the algorithm results on all four networks, for all subgraphs, converge to the true concentrations. **A.** Transcriptional regulation network of *E. coli*. All the four 5-node subgraphs were found as network motifs. Despite the low concentration of the subgraphs, they are accurately estimated with a small error ratio even with relatively few samples ( $10^5$ ). Total number of 5-nodes subgraphs in the network is  $1.43 \times 10^6$ . **B.** Transcriptional regulation network of yeast (*S.cerevisiae*). Three of the subgraphs (all but the bottom right subgraph) are found to be network motifs. Results of a high concentration subgraph (bottom right) also converge rapidly to the real concentration. Total number of 5-nodes subgraphs in the network is  $2.5 \times 10^6$ . **C.** Neuronal network of *C. elegans*. All the four 4-node subgraphs were found as network motifs. This network is characterized by relative high density (average degree = 15.5). Total number of 4-node subgraphs is  $8.75 \times 10^5$ . **D.** Ythan food web. All the four 5-node subgraphs were detected as network motifs. Total number of 5-node subgraphs is  $9.4 \times 10^5$ .

## Appendix 1: Several notes related to the algorithm

**1.1** In order to efficiently maintain the candidate edges list in the sampling process, we keep two global data structures: 1) A mapping matrix of all the edges in the network to edges indexes. 2) A global array of the largest hub edges. This is a binary array of size E (E is the number of edges in the network), where only the hub edges have value 1 in the appropriate indexes.

Whenever a hub edge is picked in the sampling process, we use the global hub edges array as a basis for the candidate edge array, and operate all the required operations on this array. Such Implementation reduces the complexity of maintaining the candidate edges lists per sample from  $O(Dn)$  to  $O(n^2)$  on a cost of  $O(E)$  additional memory.

**1.2** Note that in principle the algorithm could be made more efficient by avoiding repeated sampling of the same subgraph. In practice, however, the number of samples is much smaller than the total number of subgraphs and thus the added efficiency is small.

**1.3** In the present study, unlike (Milo 2002), the randomized networks used to detect n-node motifs were not constrained to have the same number of (n-1)-node subgraphs as the real network.

## Appendix 2: Deciding the number of samples by convergence

We used an approach similar to adaptive sampling described by (Chaudhuri 1998).

Let  $V_i = (\hat{c}_1^i, \hat{c}_2^i, \dots, \hat{c}_k^i)$  and  $V_{i-1} = (\hat{c}_1^{i-1}, \hat{c}_2^{i-1}, \dots, \hat{c}_k^{i-1})$  be the vector of estimated subgraphs concentration after iteration i and iteration i-1 respectively. We define the average instantaneous convergence rate as

$$CG_{avg} = \frac{1}{k} \sum_{j=1}^k \frac{|\hat{c}_j^i - \hat{c}_j^{i-1}|}{0.5(\hat{c}_j^i + \hat{c}_j^{i-1})} (\forall \hat{c}_j^i > C_{min})$$

and the maximal instantaneous convergence rate as

$$CG_{max} = MAX_j \left\{ \frac{|\hat{c}_j^i - \hat{c}_j^{i-1}|}{0.5(\hat{c}_j^i + \hat{c}_j^{i-1})} \mid \forall \hat{c}_j^i > C_{min} \right\}.$$

By setting the thresholds of  $CG_{avg}$ ,  $CG_{max}$  and the value of  $C_{min}$  we can adjust the required accuracy of the results and the minimum concentration of subgraphs we are interested in. Clearly there is a tradeoff between the accuracy and the required number of samples. We begin with a small number of samples and at each iteration we increase the number of samples and merge the results. We repeat the iterations until we get a small enough difference in the concentrations of all subgraphs between the current iteration and the previous one.

An alternative way for evaluating the quality of the results is to observe each subgraph type result separately. For each subgraph we can get an idea about the confidence of the estimation by its convergence rate and its number of hits (the number of samples a specific subgraph type was sampled).

## Appendix 3: Network databases

(N=number of nodes, E=number of edges). Self edges were excluded. Transcription network of E. coli (Shen-Orr 2002), version 1.1 (N=423, E=519) available at <http://www.weizmann.ac.il/mcb/UriAlon/>. Transcription network of yeast (*S. cerevisiae*) (Milo 2002), version 1.3 (N=685, E=1052) available at

<http://www.weizmann.ac.il/mcb/UriAlon/> was based on selected data from (Costanzo 2001; Milo 2002). Neuronal synaptic connection network of *C. elegans* (N=280, E=2170) was based on (White 1986) as arranged in (Achacoso and Yamamoto 1992). The network was compiled with a cutoff of one synapse for connections between neurons. Target muscle cells were excluded. WWW network of hyperlinks between web pages in ndu domain (N=3.25x10<sup>5</sup> , E=1.46x10<sup>6</sup>) (Barabasi and Albert 1999). Food web of birds, fishes and invertebrates, Ythan Estuary (N=83,E=391) (Williams and Martinez 2000).