Efficient sampling algorithm for estimating subgraph concentrations and detecting network motifs

N. Kashtan[1,3], S. Itzkovitz[1,2], R. Milo[1,2] and U. Alon[1,2,]*

[1]Department of Molecular Cell biology, [2]Department of Physics of Complex Systems and [3]Department of Computer Science and Applied Mathematics, Weizmann Institute of Science, Rehovot 76100, Israel

ABSTRACT

Summary: Biological and engineered networks have recently been shown to display network motifs: a small set of characteristic patterns that 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 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 runtime 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 exhaustive enumeration algorithms. We present results for high-order motifs in several biological networks.

Availability: A software tool for estimating subgraph concentrations and detecting network motifs (mfinder 1.1) and further information is available at http://www.weizmann.ac.il/mcb/UriAlon/

Contact: urialon@weizmann.ac.il

INTRODUCTION

Electronic devices are usually built of recurring circuit elements. Recently, it was found that biochemical and neuronal networks share a similar property: they contain recurring circuit elements called network motifs. Network motifs are subgraphs that occur in the network far more often than in randomized networks (Milo *et al*., 2002). Other types of networks such as ecological and technological networks contain different sets of characteristics network motifs

(Milo *et al.*, 2004). In the case of biological regulation networks, it has been suggested that network motifs play key information processing roles (Shen-Orr *et al*., 2002). Three major network motifs were found in the transcription network of bacteria and yeast (Milo *et al*., 2002; Shen-Orr *et al*., 2002, Lee *et al*., 2002). One of these, the feed-forward loop (FFL), 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 FFL was demonstrated experimentally using high-resolution gene expression measurements on the arabinose utilization system of *Escherichia coli* (Mangan *et al*., 2003). A second network motif in transcription networks, the single-input module, has been shown theoretically (Shen-Orr *et al*., 2002) and experimentally (Kalir *et al*., 2001; Ronen *et al*., 2002; Zaslaver *et al*., 2004) to generate temporal programs of expression. The temporal order of expression of these genes corresponds to the functional order of the gene products (Laub *et al*., 2000; Kalir *et al*., 2001; Ronen *et al*., 2002; Zaslaver *et al*., 2004; McAdams and Shapiro, 2003). Signaling networks and developmental transcription networks show these motifs, as well as other motifs (Milo *et al.*, 2004; Lahav *et al.*., 2004). More generally, network motifs raise 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 five-node subgraphs, etc.). Motifs are those subgraphs that occur significantly more often in the real network than in randomized networks. As a stringent control, the random network ensemble preserves the single-node characteristics of the real network: the number of incoming, outgoing and mutual edges for each node.

---

*[*]To whom correspondence should be addressed.

There are therefore two main tasks in detecting network motifs: (1) generating an ensemble of proper random networks (Milo *et al.*, 2003) and (2) counting the subgraphs in the real network and in 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 only certain classes of subgraphs such as cycles (Johnson, 1975; Alon *et al.*, 1997) and cliques (Akkoyunlu, 1973; Nesetril and Poljak, 1985), reviewed in Bezem and Van Leeuwen (1987). Approaches to 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 *et al.*, 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 *et al.*, 2000; Kuramochi and Karypis, 2001). An approach to approximating frequencies of subgraphs in a given non-directed, labeled graph was developed by Duke *et al.* (1995), based on the regularity lemma of graphs (Szemeredi, 1978; Alon *et al.*, 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 limited to three-node subgraphs). The runtime of the algorithm grows polynomially with network size. Thus, there is a lack of practical algorithms for counting subgraphs in large networks.

In a previous study, we developed an exhaustive-enumeration algorithm that counts all the subgraphs with a given number of nodes, *n*, in the network (Milo *et al.*, 2002; Shen-Orr *et al.*, 2002). For example, for $n = 3$, the algorithm computes the number of appearances of all the 13 types of three-node connected directed subgraphs. The performance of this algorithm 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 generate many subgraphs combinatorially (Itzkovitz *et al.*, 2003). 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 for subgraphs with $n \geq 5$.

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 enumerate subgraphs exhaustively 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 detect network motifs reliably. The sampling method is useful for analyzing very large networks or for detection of high-order motifs, which are beyond the reach of exhaustive enumeration 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 subgraphs of type *i* is N_i. The concentration of *n*-node subgraphs of type *i* is the ratio between their 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 FFL (subgraph M4 in Table 2) appears 42 times in the *E.coli* gene transcriptional network studied in Shen-Orr *et al.* (2002). The total number of three-node connected subgraphs in the network is 5206, and therefore the FFL concentration is $C_{\text{FFL}} = 42/5206 = 0.008$.

Subgraphs sampling

The algorithm samples *n*-node subgraphs by picking random connected 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 expand the subgraph iteratively 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 randomly choose an edge from the list. Finally, the sampled subgraph is defined by the set of *n* nodes and all the edges that connect between these nodes in the original network (not just the edges that were picked by the expansion process). (See algorithm formal description in Fig. 1.)

Exact correction for non-uniform sampling

A specific subgraph is a set of *n* connected nodes in the network. The probabilities of sampling different specific subgraphs in the network are not equal even if they have the same topology. In order to correct for this, we calculate the probability, *P*, of sampling a specific subgraph. Each subgraph type receives a score. 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.

To sample 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 such possible ordered sets of $n - 1$ edges [denoted as $(n - 1)$-permutations] that could lead to sampling of the subgraph.

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_S.

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 neighboring edges of E_S.

   Omit from L all edges between members of V_S. If L is empty return to 1.

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 n-node subgraph S.

5. Calculate the probability P to sample S.

Fig. 1. Sampling algorithm. Steps 1–5 represent a single sampling step; this is repeated S_T times.

Toy Network:

Probability to sample {1,2,3}:
There are 2 possibilities to sample {1,2,3}:
1. Pick first (1,2): Pr=1/E=1/6.
   then pick (1,3): Pr=1.
       Pr[(1,2) then (1,3)]=1/6*1=1/6.
2. Pick first (1,3):Pr=1/E=1/6.
   then pick (1,2): Pr=1.
       Pr[(1,3) then (1,2)]=1/6*1=1/6.
In Total: Pr[{1,2,3}] = 1/6 + 1/6 = 1/3=12/36

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

Fig. 2. Different probabilities of sampling different subgraphs. Example of a toy network with seven nodes and six directed edges. The probabilities of sampling two different three-nodes subgraphs are different, although they both are of the same subgraph type (V-shaped outgoing edges).

The probability of sampling the subgraph is the sum of the probabilities of all such possible ordered sets of $n - 1$ edges:

$$P = \sum_{\sigma \in S_m} \prod_{E_j \in \sigma} \Pr[E_j = e_j | (E_1, \ldots, E_{j-1}) = (e_1, \ldots, e_{j-1})].$$

Where S_m is a set of all $(n - 1)$-permutations of the edges from the specific subgraph edges that could lead to a sample of the subgraph. E_j is the j-th edge in a specific $(n - 1)$-permutation (σ).

In Figure 2, we illustrate this procedure on a simple toy 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 unbiased estimation of subgraph concentrations.

Calculating the concentrations of *n*-node subgraphs

We define a score S_i for each subgraph type i. Initially we set all S_is to zero. For every sample, we add the weighted

Table 1. Sampling method versus exhaustive enumeration on a WWW network

Subgraph		Exhaustive enumeration Total no. of subgraphs 287M (runtime: 2.9 h)		Sampling method No. of samples 5K (runtime: 15 s)	No. of samples 50K (runtime: 37 s)	No. of samples 2.5M (runtime: 28 min)
ID		Appearances	Concentration ($\times 10^{-3}$)	Concentration ($\times 10^{-3}$)	Concentration ($\times 10^{-3}$)	Concentration ($\times 10^{-3}$)
6		47 015 127	163.8	181.2	168.4	162.7
12		2 319 911	8.1	10.3	6.7	8.2
14		1 363 964	4.8	6.0	4.9	4.8
36		218 449 147	761.0	732.2	754.8	762.2
38*		499 763	1.74	1.97	1.75	1.73
46*		1 164 456	4.1	4.9	4.1	4.1
74		4 049 373	14.1	17.4	15.7	13.9
78		4 954 123	17.3	18.5	17.7	17.2
98		9474	0.030	0.006	0.048	0.030
102		40 607	0.14	0.08	0.16	0.14
108*		309 167	1.08	1.08	1.08	1.08
110*		106 614	0.37	0.51	0.37	0.37
238*		6 779 926	23.6	25.9	24.2	23.5

Results of the sampling method of three-node subgraphs compared with the exhaustive 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 three-node connected subgraphs appear in the network. It can be seen that as few as 5000 samples (out of 287 million three-node subgraphs) already give quite a good estimate of all the subgraph concentrations.
*Highlighted subgraphs were found to be network motifs.

score $W = 1/P$ to the accumulated score, S_i, of the relevant subgraph type i: $S_i = S_i + W$. After S_T samples, assuming we sampled L different subgraph types, we calculate the estimated subgraph concentrations

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

Runtime analyses

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

RESULTS

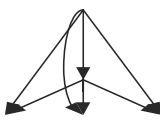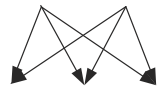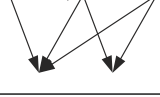### Comparing sampling method results with exhaustive enumeration

In Table 1, we show the results of the sampling method with different numbers of samples for three-node subgraphs on a WWW network (Barabasi and Albert, 1999) with 3.25×10^5 nodes, 1.46×10^6 edges. The total number of connected three-node subgraphs in the network is 2.87×10^6. Running the

algorithm with as few as 5000 samples gives a good estimation of all 13 three-node subgraph concentrations (Table 1). Even with 5000 samples, the five network motifs (highlighted with an asterisk) are detected as significant versus 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 σ_{rand} are the mean and SD in the randomized networks] (Milo *et al.*, 2002). The runtime was about 500 times faster than with the exhaustive 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 three-node subgraphs that appear in the transcription network of *E.coli* (Shen-Orr *et al.*, 2002) as well as the four and five-node subgraphs that are network motifs. It can be seen that the sampling method estimates the subgraph concentration very accurately even for subgraphs with a relatively low concentration (e.g. five-node motifs with $C = 10^{-5}$).

Generally, we find that in a variety of networks, network motifs have relatively high concentrations. Most three and four motifs of size 3–4 have $C_i > 10^{-3}$; five-node motifs usually have $C_i > 10^{-5}$. This suggests that the

Table 2. Subgraphs of size 3–5 in the transcriptional regulation network of *E.coli*

Subgraph size	Subgraph ID	Shape	Full enumeration Appearances (Z-score)	Concentration ($\times 10^{-3}$)	Sampling method Concentration ($\times 10^{-3}$) (Z-score)	No. of samples
3	S1		4777	917.60	916.60	1K (~5K total three-node subgraphs)
	S2		160	30.73	31.13	
	S3		227	43.60	43.64	
	M4		42 ($z = 10$)	8.07	8.69 ($z = 10$)	
4	M5		209 ($z = 9$)	2.49	2.69 ($z = 8$)	10K (~85K total four-node subgraphs)
	M6		51 ($z = 15$)	0.61	0.65 ($z = 15$)	
5	M7		54 ($z = 120$)	0.038	0.035 ($z = 30$)	50K (~1.4M total five-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 ($z = 8$)	

Results of the sampling method versus exhaustive enumeration for subgraphs size of 3–5 in the transcription network of *E.coli* (Shen-Orr *et al*., 2002). For size $n = 4$ and 5, only motifs are shown. Statistical significance is represented by the Z-score [$Z = (C_{real} - \langle C_{rand} \rangle)/\sigma_{rand}$]. It can be seen that the sampling method gives a very accurate estimation with a relatively small number of samples. Five-node subgraphs, although appearing in low concentrations, show good results with 50K samples—the total number of five-nodes subgraphs is 1.4×10^6. All the motifs detected by exhaustive enumeration were also detected by the sampling method (with $Z > 5$).

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 in maintaining 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 merging edge lists and throwing away edges that connect between nodes that were already picked. The worst complexity is $O(Dn)$ for every sample of an 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, for every edge we need to calculate the probability of sampling the next edge. In order to do this, we need to calculate the effective degree of each node (i.e. the number of edges that expand the subgraphs by one) at each step of picking the next random edge. Using 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 n-node subgraph is typically $n - 1 \leq m \leq Kn$ (K is a small
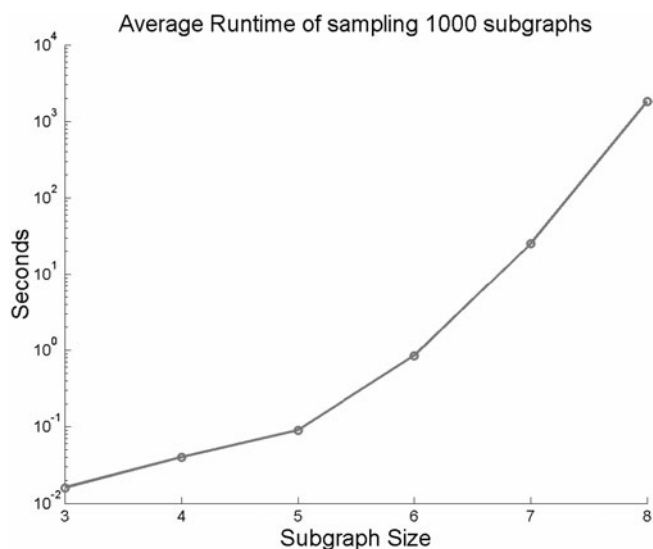
Fig. 3. Runtime per 1000 samples for different subgraph sizes: three-node up to eight-node subgraphs (semi-log scale). The network analyzed is the transcriptional regulation of *E.coli* (Shen-Orr *et al.*, 2002). The scaling of the runtime of the sampling method qualitatively agrees with the theoretical analysis of $O(K^{n-1}n^{n+1})$, where n is the subgraph size.
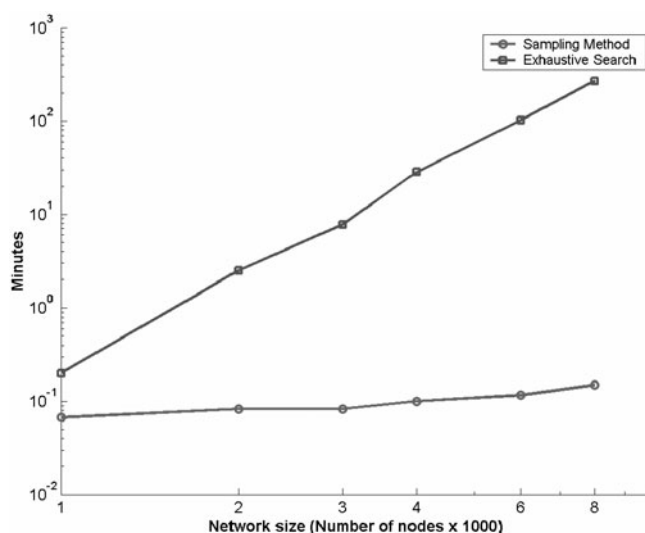

Fig. 4. Runtime of the sampling method versus an exhaustive enumeration as a function of network size (log–log scale). The networks are synthetic scale-free networks ($\gamma = 2$) with equal average connectivity ($\langle d \rangle = 2.4$). The hub degree is 10% of the total number of nodes. The sampling method was run with 100 000 samples for all the networks. The runtime of the exhaustive enumeration scales as the total number of subgraphs, while the runtime of the sampling method is almost constant.

constant which is correlated with the average degree of nodes in the network, $(n - 1)/n \leq K \leq n - 1$). Thus the number of $(n - 1)$-permutations of edges is of the order of $O(K^{n-1}n^{n-1})$. In total, we get a complexity per sample of $O(n^2) \times O(K^{n-1}n^{n-1}) = O(K^{n-1}n^{n+1})$. We conclude that the total runtime of the algorithm is $R_S = S_T \times O(K^{n-1}n^{n+1})$. This agrees qualitatively with runtime measurements for sampling subgraphs of sizes 3–8 (Fig. 3) on the transcription network of *E.coli*.

Analyzing the runtime of the sampling method versus an exhaustive enumeration

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

degree of the nodes. For the simplicity of the analyses we assume $K = 1$.

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 three-node subgraphs $r \sim 1$, for four-node subgraphs $r \sim 150$ and for five-node subgraphs $r \sim 1.5 \times 10^4$. We find that for subgraphs of four nodes and above, the runtime of the sampling method is much smaller than that of an exhaustive enumeration algorithm (Fig. 4).

For a network that does not have hubs, the ratio is

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

For such a network with $N = 10\,000$, $\langle d \rangle = 3$, $S_T = 10^5$, we find for a three-node subgraph $r \sim 0.1$, for a four-node subgraph $r \sim 0.05$ and for a five-node $r \sim 0.01$.

We conclude that for networks without hubs, the runtime of the sampling method is not smaller than the exhaustive

enumeration algorithm. However, it can be useful even in this case to run the sampling method with a small number of samples to get a low accuracy estimate of subgraph concentrations or for the purpose of detection of strong network motifs.

In order to compare the runtime of the two algorithms in practice, we generated synthetic scale-free networks (exponent $\gamma = 2$) with a varying number of nodes N, using the methods of Itzkovitz *et al.* (2003) (Fig. 4). All the networks had the same average connectivity ($\langle d \rangle = 2.4$). We set the hub degree to $D = 0.1N$. The runtime of the exhaustive enumeration 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 exhaustive 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

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 converged to the fully enumerated concentrations. Different numbers of samples were required for achieving good estimations for different subgraphs and in different networks. All of the simulations we performed, on a variety of networks, showed that the results converge toward 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 estimate reliably concentrations 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

The 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 case of networks with hubs, than exhaustive-enumeration approaches.

The sampling method is able to detect subgraphs whose concentration is very low with relatively few samples (e.g. the concentration of motifs with $c = 10^{-5}$ can be estimated accurately with only 50 000 samples, Table 2—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 of being 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 of which the other $n - 1$ nodes remain connected if the hub and its edges are removed. The second type, which we refer to as 'hub subgraphs', are all other subgraphs in the network. Hub subgraphs, which are typically dominated by many hub edges (edges touching a hub), are characterized by a small probability of being sampled. The reason for the small probability is that for every sampling we necessarily reach the hub before we complete an n-node set. Therefore the candidate edges list is large (of 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 of being sampled because there exists at least one option to sample the subgraph without reaching a hub or by 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 of sampling non-hub subgraphs is higher than that of 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 of being sampled, but this is usually compensated for by their high relative concentration. In both cases the correction for the non-uniformity makes sure that the concentration estimation is correct. A fast convergence rate is observed 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 *et al.*, 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 subgraph concentrations and in network motif detection.

Network motifs can be detected even with a relatively small number of samples

We find that network motifs can be detected even with a number of samples smaller than $1/C_i$ where C_i is the motif concentration. This is due to the fact that most motifs, especially in large networks, tended to have high Z-scores [$Z = (C_{real} - \langle C_{rand} \rangle)/\sigma_{rand}$]. 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
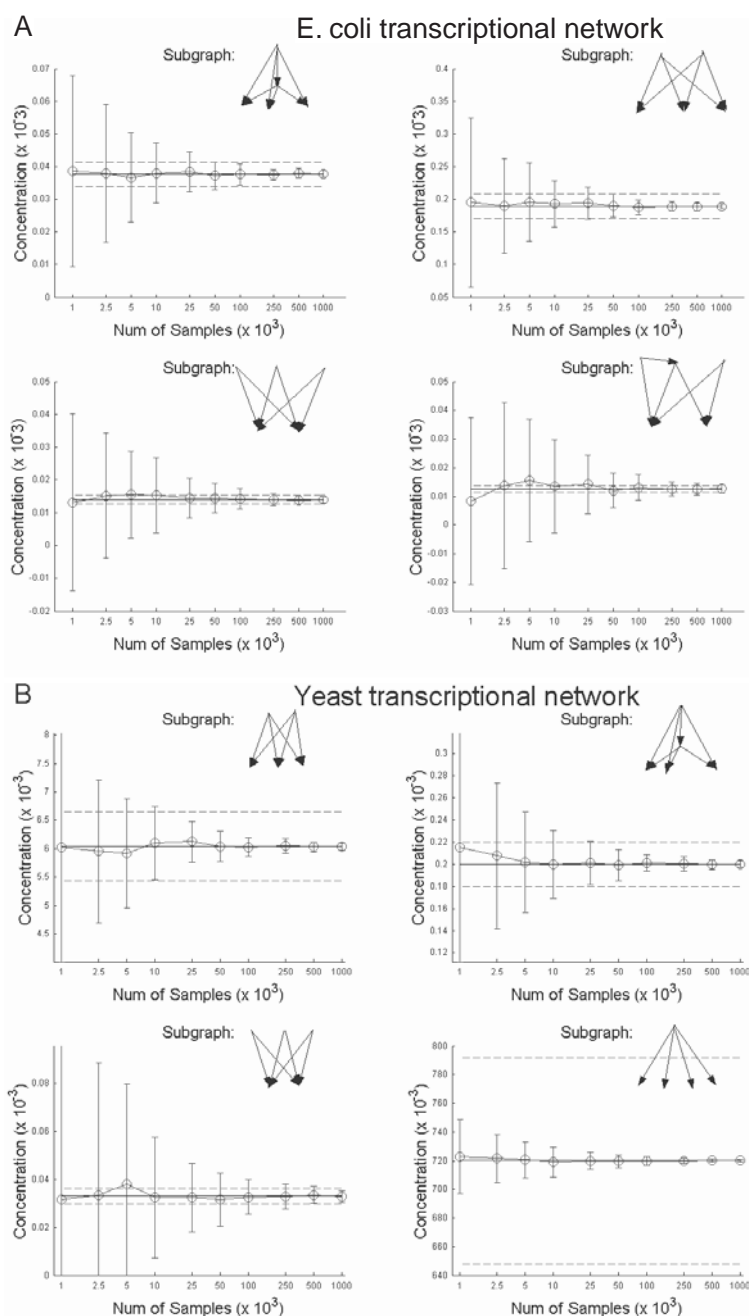
Fig. 5. Convergence of the sampling method results on different networks. Concentrations calculated by the sampling method for different subgraphs on different networks as a function of number of samples. The true concentration was found by exhaustive enumeration (horizontal full line). We ran the algorithm 100 times for each number of samples (S_T) on each of the networks. The average concentration (circles) and standard deviation are shown. Real concentrations $\pm 10\%$ are shown by dashed lines. It can be seen that the algorithm results on all four networks, for all subgraphs, converge to the true concentrations. **(A)** Transcription network of *E.coli*. All the four five-node subgraphs were found as network motifs. Despite the low concentration of the subgraphs, they are estimated accurately with a small error ratio even with relatively few samples (10^5). The total number of connected five-node subgraphs in the network is 1.43×10^6. **(B)** Transcription network of yeast (*Saccharomyces 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. The total number of connected five-nodes subgraphs in the network is 2.5×10^6. **(C)** Neuronal network of *C.elegans*. All the four four-node subgraphs were found as network motifs. This network is characterized by relative high density (average degree $= 15.5$). The total number of connected four-node subgraphs is 8.75×10^5. **(D)** Ythan Estuary food web. All the four five-node subgraphs were detected as network motifs. Total number of connected five-node subgraphs is 9.4×10^5.
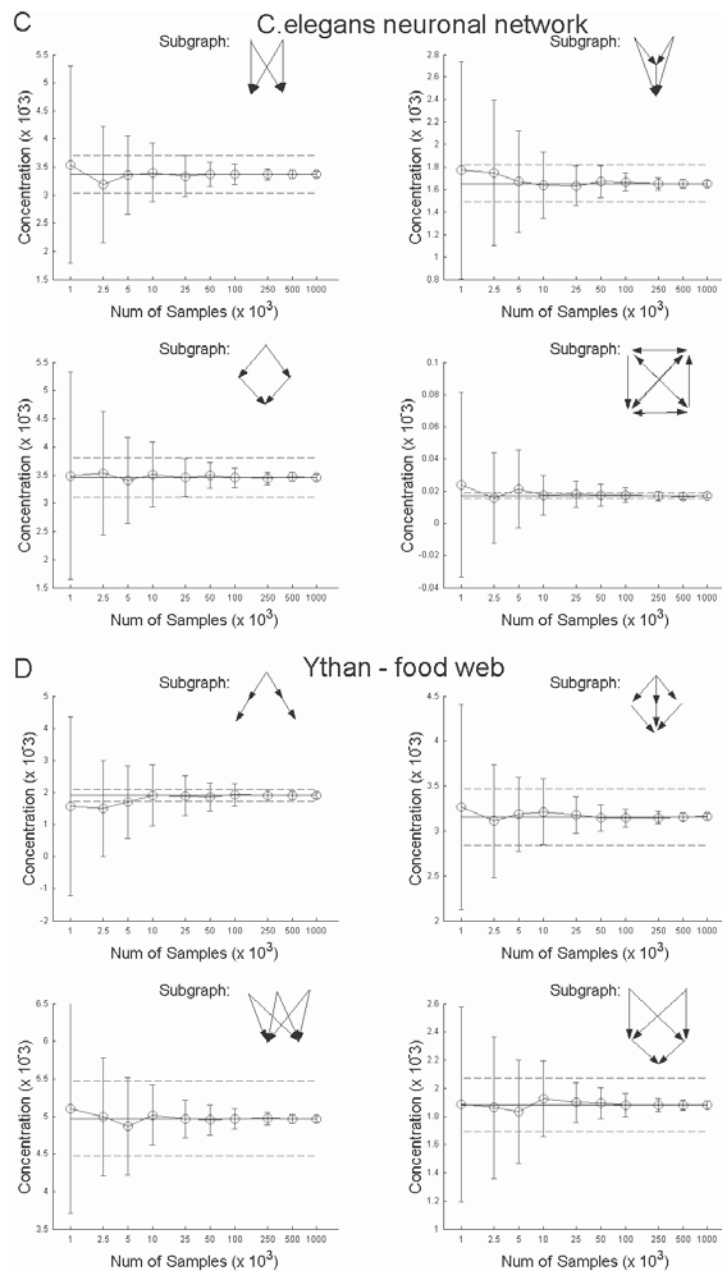
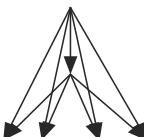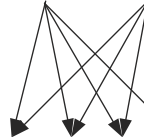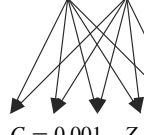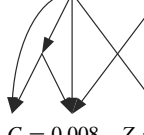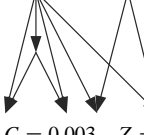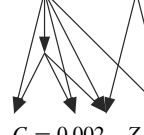**Fig. 5.** *Continued.*
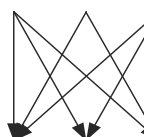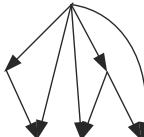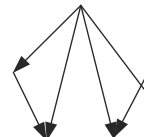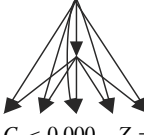
cutoff of $Z = 5$ or 10 can be used to detect significance using the sampling algorithm. Setting the Z-score cutoff to high values is important also for avoiding false positives (which can occur when $\langle C_{rand} \rangle$ is underestimated due to very low concentrations in the randomized networks) while not missing interesting motifs. The observed high Z-scores of network motifs assures us that they can be detected, even when the number of samples cannot provide a very high accuracy for the actual concentrations. In typical cases, sampling sufficient to provide 2-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 high-n motifs in networks where we have previously analyzed only $n = 3$ and 4. In the *E.coli* network, the only three-node motif is the FFL (Table 2; M4). The FFL was suggested as having a specific biological function in transcription networks. FFLs with positive regulations have been shown experimentally to function as a sign-sensitive delay element (Shen-Orr *et al.*, 2002, Mangan *et al.*, 2003). With other sign combinations, the FFL can function as a pulse-generator or response accelerator

Table 3. High-order motifs (six and seven nodes) in *E.coli* transcription network

Motifs of six and seven 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

The table summarizes the significant high-order motifs in the *E.coli* transcription network. Sampling method was run with 200 000 and 500 000 samples for detecting six-node and seven-node motifs, respectively. Detection of six-node and seven-node motifs in this network using the exhaustive enumeration algorithm was beyond reach. Concentrations ($\times 10^{-3}$) ('C') and Z-scores ('Z') of the motifs are shown. 'NA': in the random networks no appearances of this subgraph were detected, and therefore the Z-score could not be estimated.

(Mangan and Alon, 2003). At the level of four-node subgraphs, a motif appears that is a FFL with two output nodes (M6). At the level of five-node subgraphs, a FFL with three outputs appears (M7). This suggests that the proper generalization of the FFL is a motif with *n*-output nodes (Kashtan *et al.*, 2004) (Table 3; L1, L6). Similarly, the four-node bi-fan motif (M5) generalizes at the level of five-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 *et al.*, 2002). These structures function as hard-wired combinatorial decision-making circuits. Additional high-order motifs are summarized in Table 3. It can be seen that most motifs are constructed from smaller motifs following generalization rules (Table 3; L1, L6, L2, L3, L7) or by combining motifs together (Table 3; L4, L5, L8–L10). This suggests that small motifs and their generalizations can be thought of as basic building blocks of this network.

High-order motifs in the neuronal network of *Caenorhabditis 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 exhaustive enumeration algorithm we have previously detected three and four node motifs (Milo *et al.*, 2002). We applied the sampling method to the neuronal network of *C.elegans* for five- and six-node motif detection, which was beyond reach using the exhaustive enumeration algorithm. We find in this network, a different generalization form of the FFL—the multi-input FFL (Table 4, E1). This motif can act as an

integration unit of several inputs (sensory neurons) preserving the basic function of the FFL as a persistence detector. In particular it can help detect a weak signal from one input if a signal from another input has recently received (Kashtan *et al.*, 2004). We find other significant structures that are formed from combinations of three and four-node motifs (E2, E4). In addition, we find motifs (E5, E7) that are similar in structure to two-layer perceptrons (Rosenblatt, 1962) (feed-forward neural networks). Two-layer perceptrons can implement functions such as XOR (Exclusive OR), which cannot be implemented with single layer perceptrons (Hertz *et al.*, 1991). Multi-layer neurons circuit motifs (E5–E8, E11–E14) can in principle perform complex computations using suitable weights on the edges and different input functions on the nodes (Ackley *et al.*, 1985; Hertz *et al.*, 1991).

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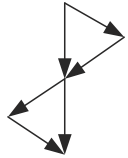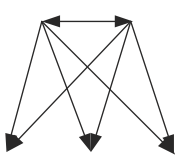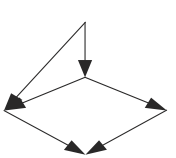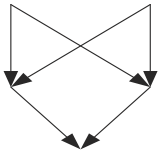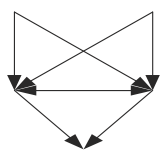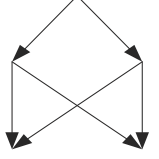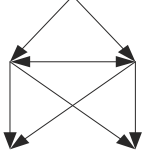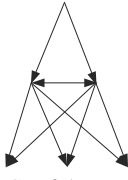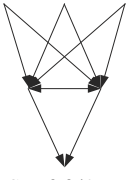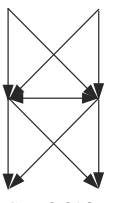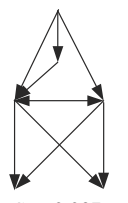
The ability to estimate the subgraph content of a network may be useful in a number of fields. For example, solving short-time diffusion or transport problems on networks (Lovasz, 1993; Bosiljka and Rodgers, 2002; Kim *et al.*, 2003) will be aided by knowledge of the local structure statistics. For motif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

ACKNOWLEDGEMENTS

Table 4. High-order motifs (five and six nodes) in the neuronal network of the nematode *C.elegans*

Motifs of five and six nodes in *C.elegans* neuronal network

5   E1

$C = 0.071$ $Z = 12$

E2

$C = 0.406$ $Z = 21$

E3

$C = 0.324$ $Z = 230$

E4

$C = 0.231$ $Z = 19$

E5

$C = 0.170$ $Z = 20$

E6

$C = 0.420$ $Z = 180$

E7

$C = 0.343$ $Z = 22$

E8

$C = 0.687$ $Z = 370$

6   E9

$C = 0.018$
$Z = \text{NA}$

E10

$C = 0.059$
$Z = \text{NA}$

E11

$C = 0.166$
$Z = 110$

E12

$C = 0.049$
$Z = \text{NA}$

E13

$C = 0.093$
$Z = 100$

E14

$C = 0.037$
$Z = \text{NA}$

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 five-node and six-node motifs in this network using the exhaustive enumeration algorithm was beyond reach. Concentrations ($\times 10^{-3}$) ('C') and Z-scores ('Z') of the motifs are shown. 'NA': in the random networks no appearances of this subgraph were detected, and therefore the Z-score could not be estimated. We note that the presented motifs are only a partial list of all the five-node and six-node motifs that were detected.

REFERENCES

Achacoso,T.B. and Yamamoto,W.S. (1992) *AY's Neuroanatomy of C. elegans for Computation*. CRC Press, Boca Roton, FL.

Ackley,D.H., Hinton,G.E. and Sejnowski,T.J. (1985) A Learning Algorithm for Boltzmann Machines. *Cognitive Science* **9**: 147–169.

Akkoyunlu,E. (1973) The enumeration of maximal cliques of large graphs. *SIAM J. Comput.*, **2**, 1–6.

Alon,N., Duke,R., Lefmann,H., Rödl,V. and Yuster,R. (1994) The algorithmic aspects of the regularity lemma. *J. Algor.*, **16**, 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. M.Sc. thesis Universiteit Utrecht, Utrecht.

Bosiljka,T. and Rodgers,G.J. (2002) Packet transport on scale free networks. *Adv. Complex Syst.*, **5**, 445–456.

Bunge,J. and Fitzpatrick,M. (1993) Estimating the number of species: a review. *J. Am. Stat. Assoc.*, **88**, 364–373.

Chaudhuri,S., Motwani,R. and Narassaya,V. (1998) Using random sampling for histogram construction: how much is enough? *Proceedings of the ACM SIGMOD International Conference on Management of Data*, Seattle, Washington, USA, 2–4 June, ACM Press, pp. 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. *et al.* (2001) YPD, PombePD and WormPD: model organism volumes of the BioKnowledge

library, an integrated resource for protein information. *Nucleic Acids Res.*, **29**, 75–79.

Duke,R., Lefmann,H. and Rödl,V. (1995) A fast approximation algorithm for computing the frequencies of subgraphs in a given graph. *SIAM J. Comput.*, **24**, 598–620.

Dyer,M., Frieze,A.M. and Jerrum,M. (1994) Approximately counting Hamilton cycles in dense graphs. *Proceedings of the 5th ACM/SIAM Symposium on Discrete Algorithms*. ACM/SIAM Press, pp. 336–343.

Flajolet,P. and Martin,G. (1985) Probabilistic counting algorithms. *J. Comput. Syst. Sci.*, **31**, 182–209.

Frieze,A. and Kannan,R. (1999) Quick approximation to matrices and applications. *Combinatorica*, **19**, 175–220.

Gibbons,P.B. (2001) Distinct sampling for highly-accurate answers to distinct values queries and event reports. *Proceedings of the 27th International Conference on Very Large Databases*. Rome, Italy. pp. 541–550.

Hertz,J., Krogh,A. and Palmer,R.G. (1991) *Introduction to the Theory of Neural Computation.* Addison-Wesley Publishing Company. Redwood, CA.

Inokuchi,A., Washio,T. and Motoda,H. (2000) An apriori-based algorithm for mining frequent substructures from graph data. *Proceedings of PKDD2000: The Fourth European Conference on Principles and Practice of Knowledge Discovery in Databases.* Lyon, Springer, France.

Itzkovitz,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, Springer, 132 pp.

Jerrum,M. and Sinclair,A. (1996) The Markov chain Monte Carlo method: an approach to approximate counting and integration. In Hochbaum, P.S. (ed.), *Approximation Algorithms for NP-hard Problems*. PWS Publishing, Boston, pp. 482–520.

Johnson,D.B. (1975) Finding all the elementary circuits of a directed graph. *SIAM J. Comput.*, **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**, 2080–2083.

Kashtan,N., Itzkovitz,S., Milo,R. and Alon,U. (2004) Topological Generalizations of network motifs arxiv: q-bio/0312019. *Phys. Rev. E.* in press.

Kim,B.J., Hong,H. and Choy,M.Y. (2003) Quantum and classical diffusion in small-world networks. *PRB*, in press.

Kuramochi,M. and Karypis,G. (2001) Frequent subgraph discovery.*Proceedings of the 1st IEEE International Conference on Data Mining (ICDM).*

Lahav,G., Rosenfeld,N., Sigal,A., Geva-Zatorsky,N., Levine,A.J., Elowitz,M.B. and Alon,U. (2004) Dynamics of the p53-Mdm2 feedback loop in individual cells. *Nat. Genet.*, **36**, 147–50.

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**, 2144–2148.

Lovasz,L. (1993) *Random Walks on Graphs: A Survey. Combinatorics Paul Erdos is Eighty*. Bolyai Society for Mathematical Studies, Vol. 2, pp. 1–46.

McAdams,H.H. and Shapiro,L. (2003) A bacterial cell-cycle regulatory network operating in time and space. *Science* **301**, 1874–1877.

Mangan,S. and Alon,U. (2003) Structure and function of the feed-forward loop network motif. *Proc. Natl Acad. Sci., USA*, **100**, 11980–11985.

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**, 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.

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.

Milo,R., Itzkovitz,S., Kashtan,N., Levitt,R., Shen-Orr,S., Ayzenshtat,I., Sheffer,M. and Alon,U. (2004) Superfamilies of evolved and designed networks. *Science* **303**(5663): 1538–42.

Minsky,M.L. and Papert,S.A. (1969) Perceptrons. Cambridge, MIT Press.

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. *Stat. Comput.*, **5**, 25–42.

Ouzounis,C. and Karp,P. (2000) Global properties of the metabolic map of *Escherichia coli. Genome Res.*, **10**, 568–576.

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., USA*, **99**, 10555–10560.

Rosenblatt,F. (1962) Principles of Neurodynamics. New York, Spartan.

Shen-Orr,S., Milo,R., Mangan,S. and Alon,U. (2002) Network motifs in the transcriptional regualtion network of *Escherichia coli. Nat. Genet.*, **31**, 64–68.

Szemeredi,E. (1978) Regular partitions of graphs. In Bermond,J.-C., Fournier,J.-C., Las Vergnas,M. and Sotteau,D. (eds), *Proceedings of the Colleque International CNRS, No. 260.* pp. 399–401.

Wagner,A. and Fell,D. (2001) The small world inside large metabolic networks. *Proc. R. Soc. Lond. B Biol. Sci.*, **268**, 1803–1810.

White,J., Southgate,E., Thomson,J. and Brenner,S. (1986) The structure of the nervous system of the nematode *Caenorhabditis elegans. Phil. Trans. R. Soc. Lond. Ser. B*, **314**, 1–340.

Williams,R.J. and Martinez,N.D. (2000) Simple rules yield complex food webs. *Nature*, **404**, 180–183.

Zaslaver,A., Mayo,A.E., Rosenberg,R., Bashkin,P., Sberro,H., Tsalyuk,M., Surette,M.G. and Alon,U. (2004) Just-in-time transcription programs in metabolic pathways. *Nat. Genet.*, **36**, 486–491.

APPENDIX 1

Several notes related to the algorithm

(1) In order to efficiently maintain the candidate edge list in the sampling process, we keep two global data structures: (1) a mapping matrix of all the edges in the network to edge indexes. (2) An 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 an implementation reduces the complexity of maintaining the candidate edge lists per sample from $O(Dn)$ to $O(n^2)$ at a cost of $O(E)$ additional memory.

(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.

(3) In the present study, unlike (Milo *et al.*, 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

Determining the number of samples by 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 *et al.*, 1998; Gibbons, 2001) and estimating statistics on a sampled population (Bunge and Fitzpatrick, 1993). It was shown to be a hard problem (Chaudhuri *et al.*, 1998). The number of samples required for good estimation with a high probability is hard to approximate when the concentration distribution is not known a priori.

To estimate the number of samples required for convergence we used an approach similar to the adaptive sampling described by Chaudhuri *et al.* (1998).

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

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

and the maximal instantaneous convergence rate as

$$\mathrm{CG}_{\mathrm{max}} = \max_j \left\{ \frac{\left| \hat{c}_j^i - \hat{c}_j^{i-1} \right|}{0.5(\hat{c}_j^i + \hat{c}_j^{i-1})} \,\middle|\, \forall \hat{c}_j^i > C_{\mathrm{min}} \right\}.$$

By setting the thresholds of $\mathrm{CG}_{\mathrm{avg}}$, $\mathrm{CG}_{\mathrm{max}}$ 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 of evaluating the quality of the results is to observe each subgraph type result separately. For each subgraph type, we can get an idea of the confidence of the estimation by its convergence rate and its number of hits (the number of times a certain 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 *et al.*, 2002), version 1.1 (N = 423, E = 519) available at http://www.weizmann.ac.il/mcb/UriAlon/. Transcription network of yeast (*S.cerevisiae*) (Milo *et al.*, 2002), version 1.3 (N = 685, E = 1052) available at http://www.weizmann.ac.il/mcb/UriAlon/ was based on selected data from Costanzo *et al.* (2001) and Milo *et al.* (2002). Neuronal synaptic connection network of *C.elegans* (N = 280, E = 2170) was based on White *et al.* (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 the ndu domain ($N = 3.25 \times 10^5$, $E = 1.46 \times 10^6$) (Barabasi and Albert, 1999). Food web of birds, fishes and invertebrates, Ythan Estuary (N = 83, E = 391) (Williams and Martinez, 2000).