

Analysis of Biological Networks: Network Motifs*

Lecturer: Roded Sharan

Scribe: Anat Halpern and Efrat Mashiach

Lecture 4, November 16, 2006

1 Introduction

This lecture describes methods for analyzing networks in terms of their motif content. *Network motifs* are defined as "recurring patterns of interactions that are significantly over-represented". The motivation for analyzing the motif content of the network lies in the basic assumption that the over-representation of a certain motif in a network indicates it has some functional importance. Thus, exploring the abundant motifs in a network may provide with novel insights regarding the functionality of these motifs in the network. Most of the notions and analyzes described here have been developed in the laboratory of Uri Alon in the Weizmann Institute. The lecture will describe the network motifs works on three levels:

1. Empirical studies on real-world networks
2. Theoretical analysis of network models
3. Motifs in the context of network evolution

2 Empirical studies on real-world networks

2.1 Transcriptional Networks

Shen-Orr *et al* [9] analyzed the motifs in a transcription network of *E. coli*, a prokaryote widely studied in biology. The transcription network contains directed transcriptional interactions from special genes called *transcription factors* to the *operons* they regulate. An *operon* is a group of contiguous genes that are transcribed into a single mRNA molecule which is later translated into several proteins. Operons are found almost exclusively in prokaryote organisms, while in eukaryotes every gene is usually transcribed into a separate mRNA molecule. The authors have compiled a database consisting of 424 operons, containing 116 transcription factors and 577 interactions.

At first, motifs were detected by exhaustive enumeration of all the possible motifs of size n for $n = 3$ and $n = 4$. The significance of each motif was assessed by comparing the number of times it appears in the network to the number of times it appears in a randomized ensemble of networks preserving the in-degree, out-degree and mutual-degree of the original network. The *mutual-degree* refers to the number of *mutual edges* the node possesses. A *mutual edge* is a bi-directional edge between two nodes (a loop of length 2).

For $n = 3$, the only significantly enriched motif was the *Feed-forward Loop (FFL)*, depicted in Figure 1. For $n = 4$ the only significant motif was the *bi-fan*, representing a pair of transcription factors regulating the same pair of operons.

*Based on a scribe by Igor Ulitsky and Daniela Raijman

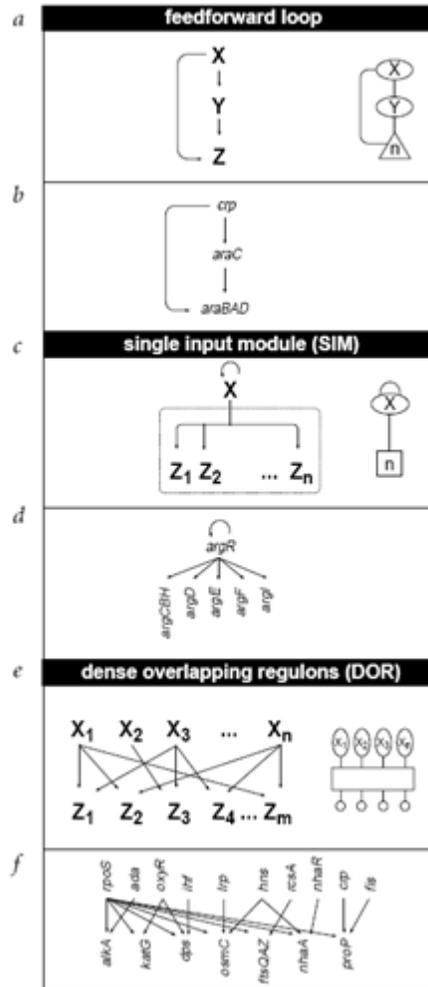


Figure 1: Source [9]. The network motifs enriched in the *E. coli* transcriptional network. **(a)** The *Feedforward Loop* motif containing two transcription factors (X - a general transcription factor and Y - a specific transcription factor, where Y is regulated by X), both regulating an effector operon Z . **(b)** A sample feedforward loop in the *L-arabinose utilization* pathway. **(c)** *Single Input Molecule* motif (SIM), built from a single transcription factor X , regulating a series of operons not regulated by any other transcription factor. The loop in X represents the auto-regulation which is usually found in those motifs. **(d)** SIM motif in the *arginine biosynthesis* pathway. **(e)** *Dense Overlapping Regulons* motif (DOR): a dense region of connections between transcription factors and operons. **(f)** a DOR motif in the *stationary phase response* pathway.

2.1.1 The Feed Forward Loop

The *Feed Forward Loop* (FFL) motif contains 2 transcription factors, X and Y , where X is referred to as the *general transcription factor* and Y as the *specific transcription factor*. The operon regulated by both X and Z is called *effector operon*. In the *E. coli*'s transcription network, 40 partially overlapping FFL have been found, encompassing 10 different general transcription factors.

In order to further establish the biological relevance of the enrichment, the authors checked the *coherence* of the FFLs in the network. A feedforward loop is *coherent* if the direct effect of X on Z has the same sign (positive or negative) as the net indirect effect of X on Z through Y . For example, if X and Y both activate

Structure	Appearances in real network	Appearances in randomized network (mean \pm s.d.)	P value
Coherent feedforward loop	34	4.4 \pm 3	$P < 0.001$
Incoherent feedforward loop	6	2.5 \pm 2	$P = 0.03$
Operons controlled by SIM (>13 operons)	68	28 \pm 7	$P < 0.01$
Pairs of operons regulated by same two transcription factors	203	57 \pm 14	$P < 0.001$
Nodes that participate in cycles*	0	0.18 \pm 0.6	$P = 0.8$

*Cycles include all loops greater than size 1 (autoregulation). P value for cycles is the probability of networks with no loops.

Figure 2: Source [9]. Summary of the significant statistical results described in Shen-Orr *et. al.* [9]. The p-values were obtained by checking the number of networks from the number ensemble which exhibited a quantity more extreme than the one found in the real network. a p-value of $P < 0.001$ designates that no random network was found with a such an extreme value. We can see in the table that the p-value of *Feedforward loop* and *Bi-Fan* are very significant.

Z and X activates Y , the loop is coherent. If, on the other hand, X represses Y , the loop is incoherent. The intuition is that if the feed forward loop carries biological importance, the number of coherent loops will be higher than expected, and the number of incoherent loops lower than expected. The results of this comparison, as shown in Figure 2, show a clear over-representation of coherent loops in the transcription network (85% of the loops found in the network are coherent). The biological reasoning for the enrichment of this motif is thought to be the ability of such machinery to render out "noise" inherent in the cellular systems. The authors have further performed a mathematical simulation, in which the effect of each gene on the others was calculated using kinetic equations. The simulations showed the theoretical advantages of this motif in filtering out accidental temporary increases in the amount of X and in fast shutdown of the expression of Z as a result of a sudden shutdown of the expression of X . See Figure 3 for simulation results.

2.1.2 Single Input Molecule and Dense Overlapping Regulons

Two motifs of variable sizes have been sought in the network. The first is the *Single Input Molecule (SIM)* motif. This motif is defined by a set of operons that are controlled by a single transcription factor (Figure 1). All the operons have to be regulated with the same sign (positive or negative) and have no additional transcriptional regulation. This motif was sought through exhaustive enumeration over all the transcription factors and 24 appearances were recorded in the entire network. The numbers of operons regulated by a SIM in the transcription network and the random network ensemble are compared in Figure 2. It is clearly seen that large SIMs are over-represented in the transcription network. The SIM motif is hypothesized to provide a detailed temporal expression program resulting from differences in the activation thresholds of different genes. Genes with a low activating thresholds will not only be activated before genes with a higher activating threshold, but also will be shutdown after them. The simulation carried out, using kinetic equations, to test this hypothesis is presented in Figure 3.

The second variable size motif sought is the *Dense Overlapping Regulons (DOR)* motif (Figure 1), which is a generalization of a *bi-fan*. This motif represents a layer of overlapping interactions between operons and a group of transcription factors, that is much more dense than corresponding structures in randomized networks. Such motifs were sought using a clustering procedure which considered all the operons regulated by two or more transcription factors. A distance measure based on the number of transcription factors regulating both operons was defined. A standard hierarchical clustering algorithm [3] was then used for combining operons into DORs. Then, additional operons regulated by the same transcription factors as the

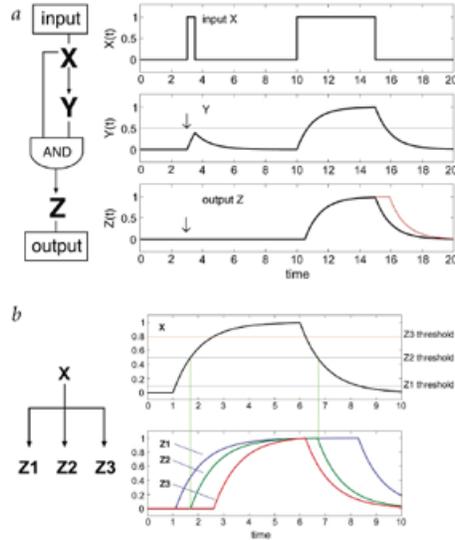


Figure 3: Source [9]. Description of the simulation providing theoretical justification for the functional use of (a) the feedforward loop and (b) the SIM motif for temporal expression programs. In the first simulation, the combinatorial regulation of the operon Z by transcription factors X and Y is modeled as a logical AND gate. The effect of X on the expression of Y and Z was modeled as a threshold function, using kinetic equations. This motif is shown to be able to ignore rapid variations in the activity of X , and therefore they don't affect the expression of Z . A short pulse in the activity of X is shown to have no effect on the expression of Z . The same effect can be accomplished by a simple cascade of $X \rightarrow Y \rightarrow Z$ (thin line in the $Z(t)$ graph), but this is shown to be theoretically inferior because of a slower shut-down. The SIM motif is shown to be able to execute a temporal program of expression through different activation thresholds of different genes. Genes with lower activation threshold are activated earliest and deactivated latest in this setting.

genes in the DOR were added to it. The exact choice of the clustering algorithm plays a role here, as the authors report different results for different algorithm choices. Shen-Orr *et al* [9] used operon clustering to derive six DORs, whose operons share common function.

The authors found that the sets of genes regulated by different transcription factors are much more overlapping than expected at random. This enrichment is quantified through the frequency of pairs of genes regulated by the same two transcription factor ($P < 0.001$, Figure 2). The large DOR motifs allow a compact modular representation of the *E. coli* transcriptional network which can be seen in Figure 4. It can be clearly seen that a single layer of DORs connects between most of the transcription factors and the effector operons. Feedforward loops and SIMs are frequent at the output of this layer.

2.2 Motifs in general networks

Following the successful application of motif extraction to the *E. coli* transcription network, a similar procedure has been applied to an ensemble of networks from highly diverse sources. Milo *et al* [8] analyzed 18 different networks from the following sources:

- Transcription networks from *E. coli* and *S. cerevisiae*.
- Synaptic connections between neurons in the nematode *C. elegans*.

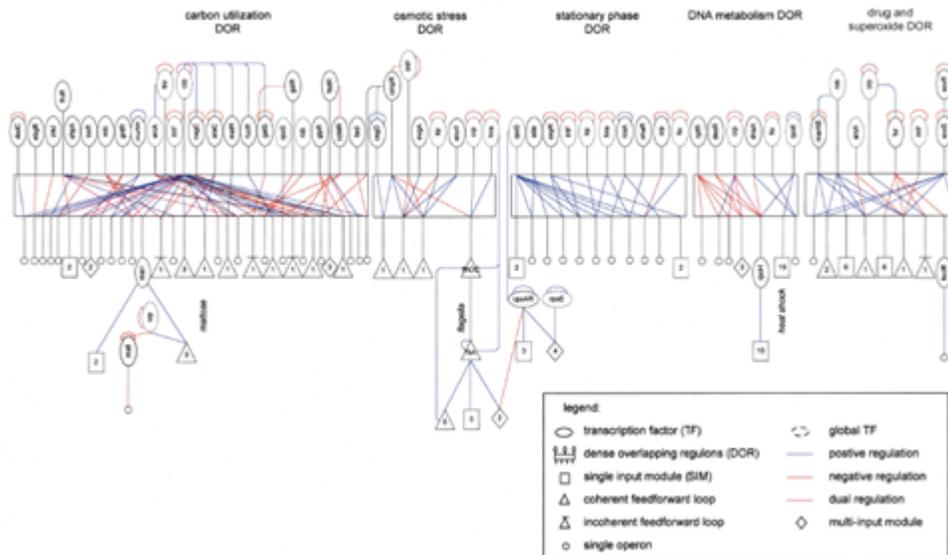


Figure 4: Source [9]. A modular representation of the *E. coli* transcription network using network motifs. Nodes represent operons and lines represent transcriptional regulations. Each DOR motif is named after the common function of its output operons. Global transcription factors regulating more than 10 operons can appear in several subgraphs.

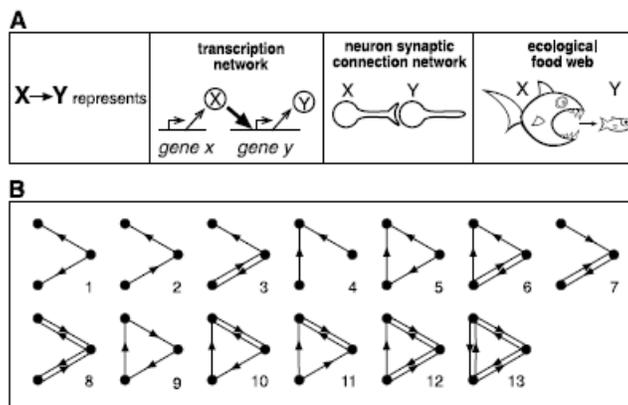


Figure 5: Source [8]. (A) The different types of networks tested by Milo *et al* [8]. (B) All the possible directed motifs for $n = 3$.

- Food webs of throphic interactions between predator and prey in different ecological systems.
- Electronic circuits.
- World Wide Web network where every web site is a node and an interaction exists between X and Y if website X is hyperlinked to website Y .

For each network, all the possible motifs for $n = 3$ (shown in Figure 5) and $n = 4$ were enumerated and compared to the average count over 1000 random networks (100 for the huge WWW network). In this case,

the randomized networks were generated while preserving the following properties of the original network:

- In-degree, out-degree and mutual degree, as before.
- The number of appearances of all $(n - 1)$ -node subgraphs. This is done to ensure that a high significance was not assigned to a pattern only because it has a highly significant sub-pattern. For example, a high number of *4-cliques* in a network is less surprising if it has an enrichment of *3-cliques*.

The randomization procedure used to ensure the above properties is based on the *Simulated Annealing* algorithm ([4], sections 10.5.1 and 10.5.3):

```

Start with the system in a known configuration, at a known energy  $E$ 
 $T = \text{temperature} = \text{hot}; \text{frozen} = \text{false}$ 
while (! frozen) {
    repeat{
        Perturb system slightly (e.g. edge-swapping)
        Compute  $\Delta E$  - the change in energy due to the perturbation
        if ( $\Delta E < 0$ )
            accept this perturbation, this is the new system configuration
        else accept with probability =  $\exp(-\Delta E/KT)$ 
    } until the system is at thermal equilibrium at this  $T$ 
    if ( $E \neq 0$ )
         $T = 0.9T$  //cool the temperature
    else frozen=true
}
return final configuration as low energy solution

```

The algorithm utilizes the edge-swapping procedure for generating random graphs, as described in Lecture 2. In addition, throughout the procedure, an energy function is used for accepting or rejecting every edge-swapping move. The energy function used is $E = \sum_k \frac{|V_{real,k} - V_{rand,k}|}{V_{real,k} + V_{rand,k}}$, where $V_{real,k}$ stands for the count of the k th $(n - 1)$ subgraph in the original network and $V_{rand,k}$ stands for the same quantity in the random subgraph. Notice that $E = 0$ if and only if the current counts of all the $n - 1$ motifs in the generated random network equal exactly the respective counts in the original network. In the simulated annealing procedure, a swap move is always accepted if it *lowers* the energy of the graph. Otherwise, it is accepted with probability $\exp(-\Delta E/KT)$ (Boltzmann Distribution), where ΔE is the difference in energy before and after the switch, T is an effective *temperature* and K is *Boltzmann's constant*. The temperature is gradually lowered during the course of the algorithm based on some preset *cooling schedule*, which is usually set for linear decay after every bulk of swaps. This procedure provides with a proper mixing of the original graph, while approximately preserving the $(n - 1)$ -motif counts.

The motifs found enriched in the different types of networks are displayed in Figure 6. The motifs were consistent across different networks from the same family. The motifs enriched in the transcription networks were shown, by Shen-Orr et al [9], to bear qualities useful for information processing. In the food webs, on the other hand, there is an under-representation of the FFL, which is common in the transcription networks. The authors suggest that this indicates that direct interaction between species at the separation of two layers (as in omnivores) are selected against in the course of evolution.

In order to test the robustness of the obtained results, the concentration of the FFL in the transcription networks has been checked in subnetworks of decreasing size of the *E. coli* transcription network (Figure 7). It can be seen that while in the random networks this concentration scales with the subnetwork size, in the real network it remains relatively constant.

Network	Nodes	Edges	N_{real}	$N_{rand} \pm SD$	Z score	N_{real}	$N_{rand} \pm SD$	Z score	N_{real}	$N_{rand} \pm SD$	Z score
Gene regulation (transcription)				Feed-forward loop			Bi-fan				
<i>E. coli</i>	424	519	40	7 ± 3	10	203	47 ± 12	13			
<i>S. cerevisiae</i> *	685	1,052	70	11 ± 4	14	1812	300 ± 40	41			
Neurons				Feed-forward loop			Bi-fan			Bi-parallel	
<i>C. elegans</i> †	252	509	125	90 ± 10	3.7	127	55 ± 13	5.3	227	35 ± 10	20
Food webs				Three chain			Bi-parallel				
Little Rock	92	984	3219	3120 ± 50	2.1	7295	2220 ± 210	25			
Ythan	63	391	1182	1020 ± 20	7.2	1357	230 ± 50	23			
St. Martin	42	205	469	450 ± 10	NS	382	130 ± 20	12			
Chesapeake	31	67	80	82 ± 4	NS	26	5 ± 2	8			
Coscheila	29	243	279	235 ± 12	3.6	181	80 ± 20	5			
Skipwith	25	189	184	150 ± 7	5.5	397	80 ± 25	13			
B. Brook	25	104	181	130 ± 7	7.4	267	30 ± 7	32			
Electronic circuits (forward logic chips)				Feed-forward loop			Bi-fan			Bi-parallel	
s15850	10,383	14,240	424	2 ± 2	285	1040	1 ± 1	1200	480	2 ± 1	335
s38584	20,717	34,204	413	10 ± 3	120	1739	6 ± 2	800	711	9 ± 2	320
s38417	23,843	33,661	612	3 ± 2	400	2404	1 ± 1	2550	531	2 ± 2	340
s9234	5,844	8,197	211	2 ± 1	140	754	1 ± 1	1050	209	1 ± 1	200
s13207	8,651	11,831	403	2 ± 1	225	4445	1 ± 1	4950	264	2 ± 1	200
Electronic circuits (digital fractional multipliers)				Three-node feedback loop			Bi-fan			Four-node feedback loop	
s208	122	189	10	1 ± 1	9	4	1 ± 1	3.8	5	1 ± 1	5
s420	252	399	20	1 ± 1	18	10	1 ± 1	10	11	1 ± 1	11
s338†	512	819	40	1 ± 1	38	22	1 ± 1	20	23	1 ± 1	25
World Wide Web				Feedback with two mutual dyads			Fully connected triad			Uplinked mutual dyad	
nd.edu6	325,729	1,46e6	1.1e5	2e3 ± 1e2	800	6.8e6	5e4 ± 4e2	15,000	1.2e6	1e4 ± 2e3	5000

Figure 6: Source [8]. Enriched network motifs found in biological and technological networks. For every family of networks, the motifs which received statistically significant score are presented along with the number of their appearances in real (N_{real}) and random (N_{rand}) networks.

2.3 Criticism of the null model

Following the publication of Milo et al [8], a critical comment has been published in *Science* magazine [1]. The authors have criticized the use of the generalized random graph as a null model for network motif detection, since it does not preserve the clustering property. For example, in the *C. elegans* synaptic connections network, neurons are spatially aggregated and connections among neurons have a tendency to form in local clusters. Two neighboring neurons have a greater chance of forming a connection than two distant neurons at opposite ends of the network. Thus, a geometrical model described in Lecture 3 is a more appropriate random model in this case. The simulation in Figure 8, carried out using a simple geometrical model network, shows that the motifs which were detected as enriched in the *C. elegans* neuronal network using a generalized random model, are over-represented in the geometrical model.

The fact that the motifs are found to be over-represented even in a random network introduces doubt into claims about their biological significance, as the observed over-representation can be explained by other properties of the network (spatial localization in this case) and not by functional importance of the motif. The authors conclude that the statistically significant motifs found in *C. elegans* are more likely to be the result of the inherently localized partitioning of the nematode's connectivity network than the result of evolutionary selection for specific motif structures.

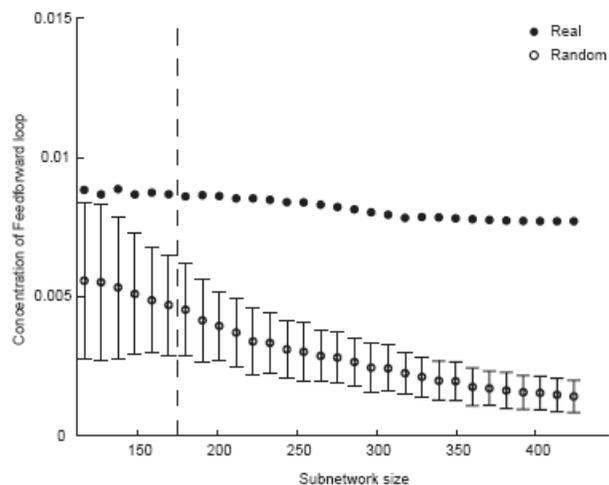


Figure 7: Source [8]. Concentration of the feedforward loop motif in both real and randomized subnetworks of the *E. coli* transcription network.

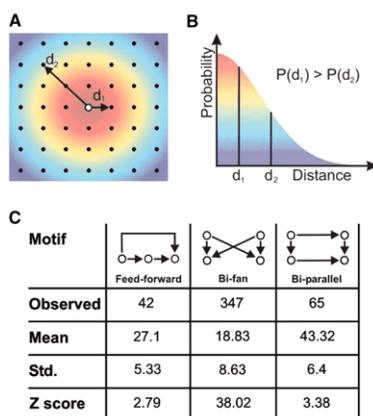


Figure 8: Source [1]. (A) Construction of a toy network using a geometrical model. A 30 by 30 grid of 900 nodes was used and the probability P of two nodes being connected reduces with the distance d between them. (B) Color-coded probability $P(d)$ of connecting to a node as a function of distance in the grid. (C) Over-representation of network motifs in the geometric network: the counts of the motifs found over-represented by Milo *et al* [8] compared with the mean number of motifs counted in 2000 randomized networks generated through edge shuffling.

3 Sampling network motifs

While the computational problem of motif finding for $n = 3$ and $n = 4$ is tractable, exhaustive enumeration becomes problematic for larger values of n , making such analysis impossible. To address this issue Kash-tan *et al* [6] have developed an algorithm for estimating the motif's concentration by subgraph sampling, with runtime asymptotically independent of the network size. This work uses the same random model as described before and focuses on the problem of counting the number of motifs in both real and random

networks. The quantity the algorithm aims to estimate is the *subgraph concentration*:

$$C_i = \frac{N_i}{\sum_j N_j}$$

where N_i is the number of occurrences of motif i . For example, if in the *E. coli* transcription network the FFL motif is found 42 times and the total number of three-node connected subgraphs is 5206, the FFL concentration is $C_{FFL} = 42/5206 = 0.008$. Instead of performing an exhaustive enumeration of all the subgraphs of the size n , the algorithm *samples* such subgraphs and estimates the frequencies of the motifs in the whole graph based on the frequencies obtained in the samples. A subgraph is sampled using a simple iterative procedure selecting connected edges until a set of n nodes is reached. This procedure is summarized in Figure 9. The algorithm keeps the current subgraph and a set of edges which can be added to the subgraph keeping it connected. At every iteration one of the edges is picked at random and added to the subgraph. At the end of the procedure, the graph that is induced from the nodes that were sampled, is returned (including edges not sampled).

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 .

Figure 9: Source [6]. Pseudo-code for sampling a single subgraph.

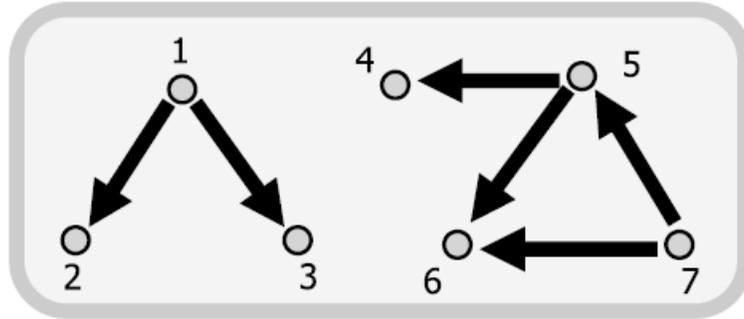
The problem with this simple approach is that the sampling is not uniform, as some subgraphs are more likely to be sampled than others (Figure 10). In order to correct this bias, a probability P is calculated, designating the probability to sample a specific subgraph. P is calculated in the following way: for every n -node subgraph, the sampling algorithm iteratively selected $n - 1$ edges. Thus, for P calculation, all the possible ordered sets of $n - 1$ edges that could lead to the sampling of the subgraph are checked (similar to the procedure shown in Figure 10):

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

Then, instead of a simple count, a score is calculated for every type of subgraph S_i . At every iteration, a weighted score $W = 1/P$ is added to S_i . The sampling procedure is repeated a large number of times. Finally, the concentration of the subgraph type is estimated based on the score it obtained:

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

Toy Network:



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

Figure 10: Source [6]. An example of a case where the algorithm samples different subgraphs with different probabilities. In this toy network, two V-shaped subgraphs ((1,2,3) and (4,5,6)) are shown to be found with different probabilities.

Subgraph	Exhaustive enumeration Total no. of subgraphs 287M (runtime: 2.9 h)	Sampling method				
		Appearances	Concentration ($\times 10^{-3}$)	No. of samples 5K (runtime: 15 s) Concentration ($\times 10^{-3}$)	No. of samples 50K (runtime: 37 s) Concentration ($\times 10^{-3}$)	No. of samples 2.5M (runtime: 28 min) Concentration ($\times 10^{-3}$)
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

Figure 11: Source [6]. Results of the sampling algorithm on three-node subgraphs compared with the exhaustive enumeration results, on the WWW network.

The results of executing the sampling algorithm on the WWW network described above are presented in Figure 11. All the 13 possible motifs are found in the network and it can be seen that 5,000 samples out of the 287×10^6 three-node subgraphs already give a good estimate of all the subgraph concentrations. Five network motifs were detected as significant due to their high scores. The runtime in this case is reduced by factor of 500 in comparison with the exhaustive enumeration algorithm.

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)	
	M7		54 (z = 120)	0.038	0.035 (z = 30)	
5	M8		271 (z = 16)	0.189	0.196 (z = 11)	50K (~1.4M total five-node subgraphs)
	M9		20 (z = 18)	0.014	0.013 (z = 8)	
	M10		18 (z = 12)	0.013	0.014 (z = 8)	
	M11					

Figure 12: Source [6]. Results of the sampling method compared to the results of the exhaustive enumeration for subgraphs with $n = 3, 4, 5$ in the transcription network of *E. coli*. The statistical significance Z-score is computed as before: $Z = (C_{real} - \langle C_{rand} \rangle) / \sigma_{rand}$.

In Figure 12 all the network motifs found in the *E. coli* transcription network for $n = 3, 4, 5$ are presented. It is shown that the sampling method estimates the subgraph concentration very accurately even for subgraphs with a relatively low concentration ($C = 10^{-5}$). The authors have also performed a theoretical analysis of the time complexity of the algorithm, showing it to be approximately $O(S_T \times K^{n-1} n^{n+1})$ where S_T is the number of iterations and K is the average degree of the nodes in the network. In Figure 13 this analysis is shown to qualitatively agree with the running time on the *E. coli* network. The runtime of the exhaustive enumeration scales with the total size of the graph, while the runtime of the sampling method is almost constant. On the other hand, the size of the subgraphs we sample does affect the runtime of the sampling method.

4 Network comparison

Comparing network structures can be a difficult task, when dealing with networks of different sizes and connectivity. An approach for comparing local topologies of different networks was presented in [7]. This approach utilizes the *significance profile* measure - a quantitative representation of the spectrum of motifs. In directed networks, the statistical significance of motif i is described by its *z-score* (z_i) in the following manner:

$$z_i = \frac{N_{real_i} - avg_i}{std_i} \quad (1)$$

where N_{real_i} is the number of the motifs observed in the network, and avg_i and std_i are the mean and standard deviation of the counts of motif i in an ensemble of random networks with the same degree sequence.

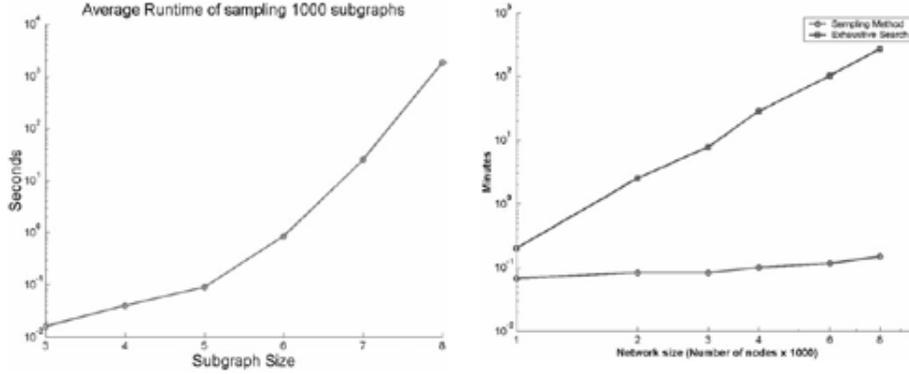


Figure 13: Source [6]. On the left, the runtime of the sampling algorithm on detection of subgraphs with $n = 3 \dots 8$. On the right, a comparison of runtimes of the exhaustive enumeration algorithm versus the sampling algorithm.

The *significance profile (SP)* is defined as:

$$SP_i = \frac{z_i}{|z|} \quad (2)$$

where $|z|$ is the vector norm. The purpose of the normalization is to discard bias resulting from network size - motifs in large networks tend to have higher z-scores than motifs in small networks.

When studying undirected networks, the profile of four-node subgraphs (tetrads) was analyzed, since only two types of three-node subgraphs exist in these networks. As the significance profiles for tetrads show a high dependency on network size, a measure called *ratio profile (RP)* was used instead:

$$RP_i = \frac{Nreal_i - avg_i}{Nreal_i + avg_i + \epsilon} \quad (3)$$

The resulting profile can be used to infer *Motif-based Superfamilies*. Figures 14 and 15 show the results for directed and undirected networks respectively.

5 Theoretical Analysis

In [5], Itzkovitz *et. al.* present a theoretical formula for computing the expected number of subgraph appearances in a network. Let us first look at a random network, where each edge exists with probability p . For a directed network G and subgraph H with n nodes and g edges, the expected number of occurrences of H in G , denoted $E(H)$, can be computed using the following formula:

$$E(H) = \lambda \binom{N}{n} p^g (1-p)^{n(n-1)-g} \sim \lambda N^n \left(\frac{d}{N}\right)^g \sim N^{n-g} \quad (4)$$

where λ is a term of order 1 which stems from the symmetry of each subgraph, and d is the average degree. The intuition for this formulation is that there are $\binom{N}{n}$ ways to choose nodes for the subgraph, and then we would like g edges to appear (probability of appearance is p), and $n(n-1) - g$ edges not to appear.

In our context, we would like to be able to calculate the expected number of appearances of a subgraph

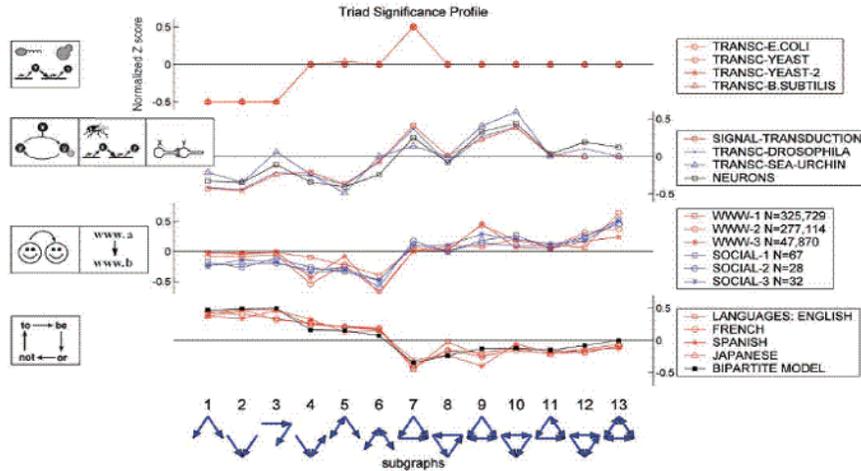


Figure 14: Source [7]. Triad Significance Profile for directed networks from various disciplines. Networks with similar characteristics are grouped into superfamilies. Networks used are: (1) *E. coli*, yeast and *B. subtilis* transcriptional networks. (2) Signal transduction networks, transcriptional networks, Neuron networks. (3) WWW and Social networks (N is the number of nodes). (4) Language networks constructed using word adjacency. Taken from [7]

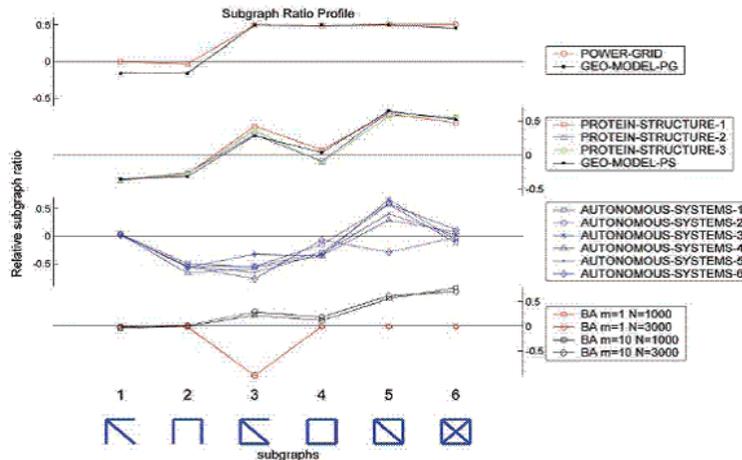


Figure 15: Source [7]. Subgraph Ratio Profile for undirected networks from various disciplines. Networks with similar characteristics are grouped into superfamilies. Networks used are: (1) Electrical power grid, geometric model (2) Secondary structure element adjacency for several large proteins (3) The Internet at the autonomous system level (4) Networks grown according to the preferential attachment BA model. m = number of edges per new node.

in a network with specific degrees, using an approximation which assumes the network is sparse (and therefore ignores non-edges). For each vertex we specify three properties: its in-degree (R_i), its out-degree (K_i), and its mutual degree (M_i) in G . Given these, we can calculate the probability of the existence of an edge from a vertex i of out-degree K_i to a vertex j of in-degree R_j : $P(\text{edge}) \approx \frac{K_i R_j}{N \langle K \rangle}$, where $\langle K \rangle$ is the average

outdegree, (equals the average indegree). However, when calculating the probability for an edge whose vertices have already been used, the degrees must be adjusted. For example, given that one edge adjacent to vertex i was used, the probability of another edge from the same vertex i to a vertex u of in-degree R_u would be $P(\text{edge}) \approx \frac{(K_i-1)R_u}{N\langle K \rangle}$. This reasoning can be extended to calculate all edge probabilities for a subgraph in a sparse generalized random network (See for example Figure 16).

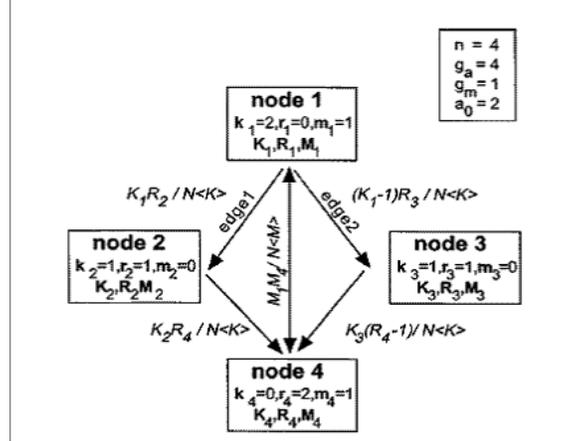


Figure 16: Source [5]. An example of a subgraph with 1 mutual edge and four single edges. k_i, r_i, m_i refer to the degrees in the subgraph. Edge probabilities are shown.

In order to calculate the mean number of appearances of a subgraph, we must take the average with respect to all possible choices of n distinct nodes $\sigma_1 \dots \sigma_n$ and multiply it by the number of possible choices of n nodes out of N . Starting with a given ordered set of nodes $\sigma = \sigma_1 \dots \sigma_n$, the probability of observing H on these nodes is calculated as follows:

$$P(H|\{\sigma\}) = \frac{N^{n-g_a-g_m}}{\langle K \rangle^{g_a} \langle M \rangle^{g_m}} \prod_{j=1}^n \binom{K_{\sigma_j}}{k_j} \binom{R_{\sigma_j}}{r_j} \binom{M_{\sigma_j}}{m_j} k_j! r_j! m_j! \quad (5)$$

where g_a is the number of single edges, g_m is the number of mutual edges, $\langle K \rangle$ is the average out-degree, and $\langle M \rangle$ is the average mutual degree.

Taking the average over all possible choices of σ , the probability of observing H is:

$$\langle H \rangle = \frac{N^{n-g_a-g_m}}{\lambda \langle K \rangle^{g_a} \langle M \rangle^{g_m}} \prod_{j=1}^n \left\langle \binom{K_i}{k_j} \binom{R_i}{r_j} \binom{M_i}{m_j} k_j! r_j! m_j! \right\rangle_{\sigma} \quad (6)$$

where λ is the number of different permutations of the nodes that result in an isomorphic subgraph. An application of the theoretical results on real data can be seen in Figure 17.

6 Evolution of motifs

6.1 Motif Conservation

In the previous section we have discussed how to find over-represented motifs in networks, and the possible biological significance of these motifs. A possible confirmation for motif importance is motif conservation.

subgraph	id	equation	transcription	neurons	www
6 *		$N(K(K-1))/2$	$1.2 \times 10^4 (-0.16\%/-0.02\%)$	$4.3 \times 10^2 (+2\%/+8\%)$	$4.7 \times 10^7 (+0.06\%/+0.5\%)$
12 *		$N(KR)$	$3.6 \times 10^2 (+0.16\%/-0.1\%)$	$8.7 \times 10^2 (+2.7\%/+3.0\%)$	$2.5 \times 10^6 (+9\%/+10\%)$
14 *		$N(KM)$	$1.9 \times 10^3 (-0.06\%/-0.06\%)$	$8.7 \times 10^4 (-0.15\%/+1.9\%)$	$3.8 \times 10^6 (-0.2\%/-0.3\%)$
36 *		$N(R(R-1))/2$	$9.6 \times 10^2 (-2\%/-0.03\%)$	$6.0 \times 10^3 (-0.4\%/+0.7\%)$	$2.2 \times 10^8 (+0.01\%/+0.1\%)$
38		$K(K-1)(RK)(R(R-1))/(K)^2$	$1.3 \times 10^1 (+1.6\%/+2.1\%)$	$1.2 \times 10^2 (+0.6\%/-28\%)$	$3.4 \times 10^5 (+0.7\%/-74\%)$
46		$(KM)^2 (R(R-1))/2 (K)^2 (M)$	$0(0\%/0\%)$	$9.3 (-10\%/-57\%)$	$8.5 \times 10^3 (-0.02\%/+8.8\%)$
74 *		$N(RM)$	$2.9 (-1.2\%/-1.8\%)$	$1.3 \times 10^2 (+1.1\%/+1.2\%)$	$4.8 \times 10^6 (-0.01\%/-0.01\%)$
78 *		$N(M(M-1))/2$	$0(0\%/0\%)$	$6.6 (-0.2\%/-0.5\%)$	$2.5 \times 10^7 (-0.4\%/-0.4\%)$
98		$(KR)^3 / 3 (K)^3$	$0(0\%/0\%)$	$4.5 (-40\%/-39\%)$	$3.3 \times 10^4 (-31\%/-26\%)$
102		$(KM)(RM)(RK)/(K)^2 (M)$	$0(0\%/0\%)$	$2 (-22\%/-15\%)$	$1.4 \times 10^2 (-11\%/-4\%)$
108		$(RM)^2 (K(K-1))/2 (K)^2 (M)$	$0(0\%/0\%)$	$1.4 (-18\%/-6\%)$	$2.9 \times 10^3 (-11\%/-44\%)$
110		$(KM)(RM)(M(M-1))/(K)(M)^2$	$0(0\%/0\%)$	$0(0\%/0\%)$	$2.3 \times 10^3 (-1.8\%/-4\%)$
238		$(M(M-1))^3 / 6 (M)^3$	$0(0\%/0\%)$	$0(0\%/0\%)$	$5 \times 10^4 (-0.04\%/-3.6\%)$

Figure 17: Source [5]. Mean numbers for 13 different subgraphs in an ensemble of random networks with a specified degree distribution. Shown are the theoretical values. Values in parentheses are percent deviations of the direct enumeration results. The left value is the percent deviation in an ensemble which allows for multiple edges, and the right value shows the deviation for an ensemble which does not allow multiple edges.

In evolution, conservation implies importance. The conservation of the proteins in a motif may be indicative of the biological importance of that motif. Wuchty *et al* [10] tested for correlation between the protein evolutionary rate and the structure of the motif it is embedded in. Motifs of size 2-5 were identified in a PPI network. If there is an evolutionary pressure to maintain specific motifs, we would expect their components to be evolutionarily conserved and have identifiable orthologs in other organisms. To test this hypothesis, the authors used a set of 678 proteins with known orthologs in 5 higher eukaryotes. The natural conservation rate indicates the fraction of the original yeast motifs that is evolutionarily fully conserved, meaning that each of their protein components belongs to a set of 678 conserved proteins. The random conservation rate is the fraction of motifs that is fully conserved for the random ortholog distribution. The conservation ratio is the ratio between the natural and random conservation rate. The results can be seen in Figure 18. The conservation rate of motif constituents was found to be tens to thousand of times higher than expected at random, suggesting conservation of motif components.

6.2 Motif Evolution

Convergent evolution is considered an indicator of optimal design. Eyes and wings are examples for convergent evolution, as they have evolved independently multiple times, despite independent origins. An interesting question to address is whether motifs, being overrepresented patterns, are the result of some optimal design, or whether they emerged through duplications of a few ancestral circuits. Given the high frequency at which genes undergo duplication, it is likely that random duplication is the process by which motifs come about. It is just as likely, however, that these patterns developed independently, and are abundant as a result of the action of natural selection. Conant and Wagner [2] showed that multiple types of transcriptional regulation circuitry in *E. Coli* and *S. Cerevisiae* have evolved independently, and not by duplication of one or a few ancestral circuits, thus indicating optimal design. In order to do so, they defined the following model: Consider a circuit topology T , which appears n times in the network. The graph G is a graph whose nodes

#	Motifs	Number of yeast motifs	Natural conservation rate	Random conservation rate	Conservation ratio
1	••	9,266	13.67%	4.63%	2.94
2	•••	167,304	4.99%	0.81%	6.15
3	•••	3,846	20.51%	1.01%	20.28
4	••••	3,649,591	0.73%	0.12%	5.87
5	••••	1,763,891	2.64%	0.18%	14.67
6	••••	9,646	6.71%	0.17%	40.44
7	••••	164,075	7.67%	0.17%	45.56
8	••••	12,423	18.68%	0.12%	157.89
9	••••	2,339	32.53%	0.08%	422.78
10	•••••	25,749	14.77%	0.05%	279.71
11	•••••	1,433	47.24%	0.02%	2,256.67

Figure 18: Source [10]. Evolutionary conservation of motif constituents. Results suggest a significant conservation of motif components.

are instances of T in the network, and whose edges connect instances that are potentially duplicates of one another, (meaning that every pair of genes are sequence-similar). Two measures were defined as indicators of common ancestry. The first measure, A , is defined as follows: $A = 1 - \frac{c}{n}$ where c is the number of components in G . The second measure, F_{max} , is defined to be the size of the largest component. The greater A is, the greater the fraction of circuits sharing a common ancestor. Figure 19 shows the two measures for different examples.

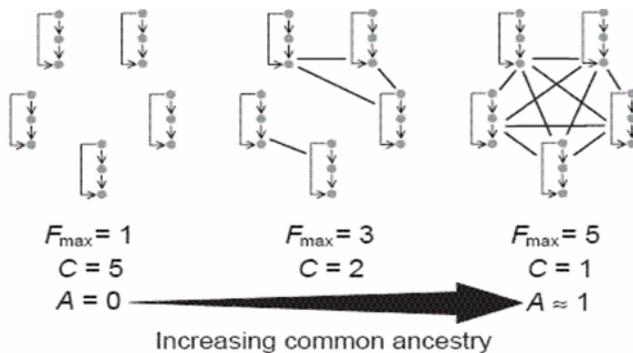


Figure 19: Source [2]. Indicators of common ancestry for gene circuits.

The large majority of the circuit types tested exhibited no significant evidence of a common ancestry, ($A = 0$ and $F_{max} = 1$). Only feed-forward loops showed marginally significant values of either A or F_{max} , ($A > 0$ or $F_{max} > 1$), but this finding is not statistically robust, as shown by permutation tests. For no circuit was A significantly different from the chance expectation, and even for feed-forward loops, most circuits showed independent ancestry. Results are summarized in Figure 20. In addition, the authors examined whether members of one gene family preferentially occurred in one type of gene circuit, which may happen if many circuits originated from one circuit. In Figure 21 we can see there is no significant evidence of such a phenomenon.

	Circuit type	Number of circuits	Number of families (C)	Index of common ancestry (A)	Largest circuit family (F_{max})
Yeast	 Feed-forward	48	44 (46.8 ± 1.9; $P = 0.08$)	0.082 (0.023 ± 0.035; $P = 0.08$)	5 (1.9 ± 1.4; $P = 0.05$)
	 Bi-fan	542	435 (469.0 ± 37.7; $P = 0.18$)	0.197 (0.135 ± 0.070; $P = 0.18$)	49 (41.0 ± 31.1; $P = 0.33$)
	 MIM-2	176	168 (164.5 ± 8.8; $P = 0.60$)	0.045 (0.065 ± 0.050; $P = 0.60$)	5 (7.4 ± 6.2; $P = 0.59$)
	 Reg. chain (3)	33	33	0	1
E. coli	 Feed-forward	11	11	0	1
	 Bi-fan	27	27	0	1

Figure 20: Source [2]. Common ancestry measures for six circuit types taken from biological networks

Organism	Circuit type	P_{motif}^a	$P_{motif/duplicate}^b$	P^c
<i>S. cerevisiae</i>	Bi-fan	0.82	0.80	NA
	Feed-forward	0.38	0.42	0.21
	Multi-input motif	0.77	0.76	NA
	Regulator chains	0.64	0.67	0.30
<i>E. coli</i>	Bi-fan	0.50	0.67	0.11
	Feed-forward	0.82	0.67	NA

Figure 21: Source [2]. Gene families are not over-represented in circuits.

References

- [1] Y. Artzy-Randrup, S.J. Fleishman, N. Ben-Tal, and L. Stone. Comment on "network motifs: Simple building blocks of complex networks" and "superfamilies of evolved and designed networks". *Bioinformatics*, 305(5687):1107, 2004.
- [2] C.G. Conant and A. Wagner. Convergent evolution of gene circuits. *Nature Genetics*, 34:264–266, 2003.
- [3] R.O. Duda and P.E. Hart. *Pattern Classification and Scene Analysis*. Wiley, New York, 1973.
- [4] W.J. Ewans and G. Grant. *Statistical Methods in Bioinformatics : An Introduction*. Springer, 2005.
- [5] S. Itzkovitz, R. Milo, N. Kashtan, G. Ziv, and U. Alon. Subgraphs in random networks. *Phys Rev E*, 68, 2003.
- [6] N. Kashtan, S. Itzkovitz, R Milo, and U. Alon. Efficient sampling algorithm for estimating subgraph concentrations and detecting network motifs. *Bioinformatics*, 20(11):1746–1758, 2004.
- [7] R. Milo, S. Itzkovitz, N. Kashtan, R. Levitt, S. Shen-Orr, I. Ayzenshtat, M. Sheffer, and U. Alon. Superfamilies of evolved and designed networks. *Science*, 303(5663):1538–1542, 2004.

- [8] R Milo, S. Shen-Orr, S. Itzkovitz, N. Kashtan, Chklovskii D., and U. Alon. Network motifs: Simple building blocks of complex networks. *Science*, 5594(298):824–827, 2002.
- [9] S. Shen-Orr, R Milo, S. Mangan, and U. Alon. Network motifs in the transcriptional regulation network of escherichia coli. *Nature Genetics*, 5594(31):64–68, 2002.
- [10] S. Wuchty, Z.N. Oltvai, and A.L. Barabasi. Evolutionary conservation of motif constituents in the yeast protein interaction network. *Nature Genetics*, 35:176–179, 2003.