Proc. 5th WABI-05, Vol. 3692 in LNBI, pp. 165-177, Springer, 2005

A Faster Algorithm for Detecting Network Motifs

Sebastian Wernicke*

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

Abstract. Motifs in a network are small connected subnetworks that occur in significantly higher frequencies than in random networks. They have recently gathered much attention as a useful concept to uncover structural design principles of complex networks. Kashtan et al. [*Bioinformatics*, 2004] proposed a sampling algorithm for efficiently performing the computationally challenging task of detecting network motifs. However, among other drawbacks, this algorithm suffers from sampling bias and is only efficient when the motifs are small (3 or 4 nodes). Based on a detailed analysis of the previous algorithm, we present a new algorithm for network motif detection which overcomes these drawbacks. Experiments on a testbed of biological networks show our algorithm to be orders of magnitude faster than previous approaches. This allows for the detection of larger motifs in bigger networks than was previously possible, facilitating deeper insight into the field.

1 Introduction

Motivation. Based on the idea that "evolution preserves modules that define specific [...] functions" [20], Milo et al. [14, 15] propose to uncover the structural design principles of biological networks[1] by detecting small subnetworks which occur in significantly higher frequencies than in random networks. These "topological modules" [20] are called *network motifs*.[2]

Some excitement has surrounded the network motif approach with the original paper by Milo et al. [15] being cited well over 40 times in some major scientific journals as of June 2005. The analysis of network motifs has led to interesting results (of which we only name a few here), e.g., in the areas of protein-protein interaction prediction [1] and hierarchical network decomposition [7]. The transcriptional network of *Escherichia Coli* displays motifs to which specific functionalities such as the generation of temporal expression programs or the response to

* Supported by Deutsche Telekom Stiftung and Studienstiftung des deutschen Volkes.

[1] We use the terms "network" and "node" for fields outside mathematics and computer science. The terms "graph" and "vertex" are used for discussing algorithmic aspects.

[2] Note that the term "network motif" has been used in other contexts as well and, e.g., may also refer to a common subnetwork in a set of given networks [17] or to any small labeled subnetwork (without considering connectivity or isomorphy) [5].

fluctuating external signals can be attributed [15, 18], suggesting that network motifs play key information processing roles in this type of network [9]. The same motifs as in the transcriptional interaction network of *E. Coli* were also identified for the yeast *Saccharomyces Cerevisiae*, possibly hinting that common network function implies the sharing of common motifs [11].

To put motif research in proper perspective, it should be noted that it has also been met with some criticism. Artzy-Randrup et al. [2] found that certain random network models lead to a display of motifs although there is no explicit selection mechanism for local structures (Milo et al. answer this criticism in [13]). Vázquez et al. [19] demonstrated that global network features such as the clustering coefficient also influence local features such as the abundance of certain subgraphs.

Previous Work. Much work related to network motifs has been spent on interpreting and applying the general concept, but considerably less on the involved algorithmics. Finding network motifs consists of three subtasks:

1. Find which subgraphs occur in the input graph (and in which number).
2. Determine which of these subgraphs are topologically equivalent (i.e., isomorphic) and group them into subgraph classes accordingly.
3. Determine which subgraph classes are displayed at a much higher frequency than in random graphs (under a specified random graph model).

Performing the first subtask by explicitly enumerating all subgraphs of a certain size can be time consuming due to their potentially large number even in small, sparse networks. For this reason, Kashtan et al. [9] propose an algorithm that estimates subgraph occurrences from a randomly sampled set of subgraphs. We discuss this algorithm in full detail in Section 2, mentioning only in passing here that it provides only biased sampling. This leads to considerable drawbacks such as an inconsistent sampling quality and the need for a computationally expensive bias correction. Besides [9], we are only aware of the work by Duke et al. [6] on approximating the number of size-k subgraphs in a given graph. Their algorithm, however, has no practical relevance since the input graph has to be astronomically large (as compared to k) in order to ensure a reasonable quality of approximation.

Much work has already been done concerning the second subtask and we rely on the *nauty* algorithm [12] for performing it in practice.

As to the third subtask, the standard approach for determining subgraph significance so far has been to explicitly generate an ensemble of random graphs under a given random graph model. One popular of these random graph models—which we also focus on in this work—is that of random graphs which preserve the degree sequence of the original graph. (Alternative choices, e.g., additionally preserve the number of bidirectional edges.) While there has been some research concerning the properties of graphs with prescribed degree sequence (such as the average path length [16]), the problem of subgraph distribution within such graphs has only been studied for directed sparse random graphs with *expected* degree sequences [8].

Contribution and Structure of this Work. We give significant improvements for the first and third subtask of motif detection. Based on a comprehensive analysis of the drawbacks encountered when using the subgraph sampling approach proposed by Kashtan et al. [9], Section 2 presents a new algorithm for subgraph sampling which does not suffer from these drawbacks (and has some additional useful features). While this comes at the price of only being able to control the *expected* number of samples, our proposed algorithm is much easier to implement and experiments in Section 4 reveal it to be orders of magnitude faster than the algorithm of Kashtan et al. As to the task of determining subgraph significance, Section 3 proposes a new approach that does not require the explicit generation of random graphs with a prescribed degree sequence. This approach leads to a faster algortithm that is moreover able to focus on determining the significance of specific subgraphs (which is not possible with previous approaches).

The proposed new algorithms have been implemented in C++, the source code is freely available online at http://www.minet.uni-jena.de/~wernicke/motifs/. We show in Section 4 that in a testbed of biological networks, our algorithm detects network motifs significantly faster than the implementation of Kashtan et al. This enables the analysis of larger networks and more complex motifs than previously possible.

2 A Faster Algorithm for Subgraph Sampling

Introduction. The algorithm for subgraph sampling suggested by Kashtan et al. [9] is based on the idea that we start by selecting a random edge in the input graph and then randomly extend this subgraph until we obtain a connected subgraph with the desired number of vertices. Subsection 2.1 discusses this approach and its main drawbacks. We present a new approach to subgraph sampling (which is based on randomized enumeration) in Subsection 2.2. Note that, due to lack of space, we omit the proofs of the theorems and lemmas presented in this section.

Notation. Basic familiarity with graph-theoretic terminology is assumed. Given a graph $G = (V, E)$ (which can be directed), we let $n := |V|$ and assume that all vertices in V are uniquely labeled by the integers $1, \ldots, n$. We write "$u > v$" to abbreviate "label(u) > label(v)." For a set $V' \subseteq V$ of vertices, its neighborhood $N(V')$ is the set of all vertices from $V \setminus V'$ which are adjacent to at least one vertex in V'.

A connected subgraph that is induced by a vertex set of cardinality k is called *size-k subgraph*. For a given integer k, the set of all size-k subgraphs in G can be partitioned into sets $\mathcal{S}_k^i(G)$ called *subgraph classes* where two size-k subgraphs belong to the same subgraph class if and only if they are isomorphic. The *concentration* $\mathcal{C}_k^i(G)$ of a subgraph class $\mathcal{S}_k^i(G)$ is defined as $\mathcal{C}_k^i(G) := |\mathcal{S}_k^i(G)| \cdot (\sum_j |\mathcal{S}_k^j(G)|)^{-1}$. For a graph G, an integer k, and a set $\mathcal{R}$ of size-k subgraphs that were randomly sampled in G by an algorithm $\mathcal{A}$, a mapping $\hat{\mathcal{C}}_k^i : (\mathcal{R}, G) \to [0, 1]$ is called an *estimator* for $\mathcal{C}_k^i(G)$. We say that $\hat{\mathcal{C}}_k^i(\mathcal{R}, G)$

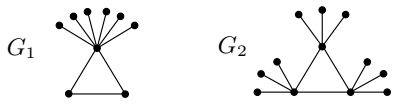

Fig. 1. Graphs G_1 and G_2 have an equal number of (connected) size-3 subgraphs. The subgraph $\triangle$ occurs exactly once in each of them. As outlined in the text, ESA oversamples the subgraph $\triangle$ in both G_1 and G_2. The oversampling is worse for G_1.

is *unbiased* (with respect to $\mathcal{A}$) if the expected value of $\hat{\mathcal{C}}_k^i(\mathcal{R}, G)$ equals $\mathcal{C}_k^i(G)$ and *biased* otherwise.

2.1 The Previous Approach: Edge Sampling

For a given graph $G = (V, E)$ and an integer $k \geq 3$, Kashtan et al. [9] suggest to sample a random subgraph by starting with a randomly chosen edge and then adding neighboring vertices until a subgraph of the desired size k is obtained:

Algorithm: Edge Sampling(G, k) (ESA)
Input: A graph $G = (V, E)$ and an integer $2 \leq k \leq |V|$.
Output: Vertices of a randomly chosen size-k subgraph in G.

01 $\{u, v\} \leftarrow$ random edge from E
02 $V' \leftarrow \{u, v\}$
03 **while** $|V'| \neq k$ **do**
04 $\quad \{u, v\} \leftarrow$ random edge from $V' \times N(V')$
05 $\quad V' \leftarrow V' \cup \{u\} \cup \{v\}$
06 **return** V'

As already noted in [9], ESA has a bias for sampling certain subgraphs more often than others. Figure 1 shows a concrete example we have constructed to illustrate this. The total number of connected size-3 subgraphs both in G_1 and G_2 is 28. Since the subgraph $\triangle$ occurs exactly once each in G_1 and G_2, we should expect that ESA samples $\triangle$ with probability $\frac{1}{28}$ within both graphs. However, Pr[ESA samples $\triangle$ in G_1] $= \frac{1}{9} \cdot 1 + \frac{2}{9} \cdot \frac{2}{8} = \frac{1}{6}$ and Pr[ESA samples $\triangle$ in G_2] $= \frac{3}{12} \cdot \frac{2}{8} = \frac{1}{16}$. This illustrates some crucial problems of ESA: The subgraph $\triangle$ is oversampled and—as a direct consequence—the only other occurring size-3 subgraph $\vee$ is undersampled. The oversampling of $\triangle$ is worse for G_1 than it is for G_2 and it is possible to show (using an adaption of the above example) that the magnitude of the oversampling cannot be estimated simply from the number of edges neighboring the oversampled subgraph. Given a set $\mathcal{R}$ of size-k subgraphs that were randomly sampled using ESA, the demonstrated bias can be overcome by using the following (unbiased) estimator [9]:

$$\hat{\mathcal{C}}_k^i(\mathcal{R}, G) := \frac{\sum_{\{G' \in \mathcal{R} \mid G' \in \mathcal{S}_k^i(G)\}} (\Pr[G' \text{ is sampled by ESA}])^{-1}}{\sum_{G' \in \mathcal{R}} (\Pr[G' \text{ is sampled by ESA}])^{-1}}. \qquad (1)$$

The main idea here is that each subgraph is (ex post facto) scored inversely proportional to the probability that ESA samples it. While it is possible to correctly estimate $\mathcal{C}_k^i(G)$ in this way, several disadvantages remain:

- The bias itself remains. E.g., subgraphs which appear in low concentration and are at the same time undersampled by ESA are hardly ever found.[3]
- Computing (1) is expensive since the calculation of *each single* probability can require as much as $\mathcal{O}(k^k)$ time [9].
- We have no estimate as to what *fraction* of subgraphs has been sampled.
- ESA can sample the same subgraph multiple times.

In the next subsection we suggest a new approach to subgraph sampling that overcomes these problems.

2.2 The New Approach: Randomized Enumeration

The idea here is to start with an algorithm that efficiently enumerates all size-k subgraphs. This algorithm is then modified to randomly "skip" some of these subgraphs during its execution, yielding an unbiased subgraph sampling algorithm.

Enumerating all size-k subgraphs. Given a graph $G = (V, E)$, the following algorithm enumerates all of its size-k subgraphs (with $N_{excl}(v, V') := N(\{v\}) \backslash N(V')$ being the *exclusive neighborhood of* v with respect to $V' \subseteq V$):

Algorithm: EnumerateSubgraphs(G, k) (ESU)
Input: A graph $G = (V, E)$ and an integer $1 \leq k \leq |V|$.
Output: All size-k subgraphs in G.

01 **for** each vertex $v \in V$ **do**
02 $\quad V_{Extension} \leftarrow \{u \in N(\{v\}) \mid u > v\}$
03 $\quad$ **call** ExtendSubgraph$(\{v\}, V_{Extension}, v)$
04 **endfor**

ExtendSubgraph$(V_{Subgraph}, V_{Extension}, v)$
E1 **if** $|V_{Subgraph}| = k$ **then output** $G[V_{Subgraph}]$ and **return**
E2 **while** $V_{Extension} \neq \emptyset$ **do**
E3 $\quad$ Remove an arbitrarily chosen vertex w from $V_{Extension}$
E4 $\quad V'_{Extension} \leftarrow V_{Extension} \cup \{u \in N_{excl}(w, V_{Subgraph}) \mid u > v\}$
E5 $\quad$ **call** ExtendSubgraph$(V_{Subgraph} \cup \{w\}, V'_{Extension}, v)$
E6 **return**

The basic idea of ESU is that—starting with a vertex v from the input graph—we add only those vertices to the $V_{Extension}$ set that have two properties: Their label must be larger than that of v and they must not be neighbor to a vertex in $V_{Subgraph}$ (other than the newly added vertex w). Some more insight into the structure of ESU can be gained by the following visualization.

[3] Kashtan et al. [9] observe that ESA can accurately estimate the concentration of $\mathcal{S}^i_k(G)$ with less than $(\mathcal{C}^i_k(G))^{-1}$ samples for subgraphs which are oversampled. In return however, other subgraphs might be missed completely for far more than $(\mathcal{C}^i_k(G))^{-1}$ samples and would consistently be overlooked as motif candidates.

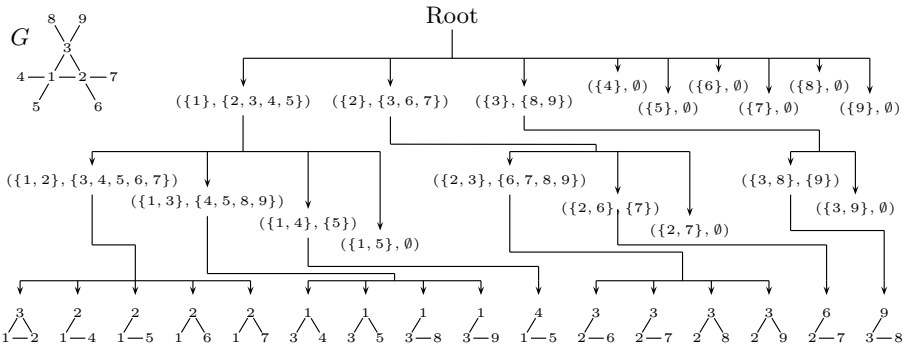

Fig. 2. The above ESU-tree corresponds to calling EnumerateSubgraphs$(G, 3)$. The tree has 16 leafs which correspond to the 16 size-3 subgraphs in G.

Definition 1. *With a call to* EnumerateSubgraphs(G, k)*, we associate a tree-graph called* ESU-tree *which represents the recursive function calls. The root at depth 0 represents the call of* EnumerateSubgraphs(G, k)*. Each call of* ExtendSubgraph$(V_{Subgraph}, V_{Extension}, v)$ *is represented by an edge from the vertex representing the caller function to a vertex representing the callee. The callee vertex is labeled* $(V_{Subgraph}, V_{Extension})$ *and located at depth* $|V_{Subgraph}|$*.*

The structure of the tree is illustrated in an example in Figure 2. Omitting the proof here, it is also the basis to establish the correctness of the ESU algorithm.

Theorem 2. *Given a graph* G *and* $k \geq 2$*,* ESU *enumerates all size-*k *subgraphs in* G *(each size-*k *subgraph is output exactly once).* □

The tree structure to represent ESU exposes some useful properties. E.g., using a technique by Knuth [10], we can randomly explore paths in the tree in order to quickly estimate the total number of size-k subgraphs in the input graph. Probably the most important feature of the ESU-tree, however, is that we can use it to efficiently sample subgraphs uniform at random (i.e., without bias).

Uniformly sampling size-k subgraphs. The ESU algorithm completely traverses its corresponding ESU-tree. Where complete traversal is too time-expensive, we can explore only parts of the ESU-tree such that each leaf is reached with equal probability. For this purpose, a probability $0 < p_d \leq 1$ is introduced for each depth $1 \leq d \leq k$ in the tree. With p_d, we determine for each child vertex at depth d whether we traverse the subtree rooted at it. This is implemented by replacing line *03* of the ESU algorithm with "With probability p_1, call ExtendSubgraph$(\dots)$" and line *E5* with "With probability p_d, call ExtendSubgraph$(\dots)$" (where $d := |V_{Subgraph}|+1$).[4] We call this new algorithm RAND-ESU.

[4] In order to reduce the sampling variance, the following more sophisticated method may be used: For a tree vertex at depth d with x children, randomly choose x' of the x children (where $x' = \lfloor x \cdot p_d \rfloor$ with probability $1 - (x \cdot p_d - \lfloor x \cdot p_d \rfloor)$ and

(To simplify the discussion, we will also use this name when all p_d are set to 1, in which case RAND-ESU is equivalent to ESU.) RAND-ESU visits each leaf of the ESU-tree with equal probability and hence estimating subgraph concentrations from its output is straightforward (the proofs are omitted).

Lemma 3. RAND-ESU *visits each leaf in the* ESU*-tree with probability* $\prod_d p_d$. □

Theorem 4. *Given a graph* G*, an integer* k*, and* $0 < p_d \leq 1$ *for* $1 \leq d \leq k$*. Let* $\mathcal{R}$ *be a set of size-*k *subgraphs obtained by running* RAND-ESU *on* G *using the probabilities* p_d*. Then,* $\hat{\mathcal{C}}_k^i(\mathcal{R}, G) := |\{G' \in \mathcal{R} \mid G' \in \mathcal{S}_k^i(G)\}| \;/\; |\mathcal{R}|$ *is an unbiased estimator for* $\mathcal{C}_k^i(G)$. □

It remains to discuss how the values p_d should be chosen. If we wish to sample an expected fraction $0 < q < 1$ of all size-k subgraphs using RAND-ESU, we have to ensure that $\prod_{1 \leq d \leq k} p_d = q$ (we omit a rigorous proof of this here). However, this still leaves us to choose the individual values, i.e., do we uniformly set every p_d equal to $\sqrt[k]{q}$ or are there better choices? Some observations are:

- Choosing whether or not to explore a subtree whose root is close to the root of the ESU-tree generally has a higher influence on the total number of explored leafs than for a subtree whose root is farther from it.
- The parameters p_d influence the distribution of the sampling, i.e., if p_d is small for small d, some local neighborhoods in the input graph are likely not to be explored at all while others will be explored extensively.
- The running time is influenced from an amortized point of view: If the p_d values are large for small values of d (and hence small for larger d), much of the ESU-tree is explored but only comparably few leafs are reached.

As a general rule from these observations, the parameters p_d should be larger for small d and become smaller as d increases—as long as the sacrifice made with respect to the amortized running time per sample is acceptable. This ensures a lower variance for the number of samples and the exploration of many different regions in the input graph.

Concluding this section, while RAND-ESU—as compared to ESA—requires a choice of sampling parameters and only allows for controlling the expected number of samples, it has a lot to offer in return. Most importantly it is unbiased, which rules out the respective disadvantages of ESA. Also, it is much faster (see Section 4) and easier to implement since we do not require any bias-correcting parts. Contrary to ESA, our new algorithm never samples more subgraphs than the input graph contains and results become exact as the number of samples reaches the total number of size-k subgraphs in the input graph.

3 Direct Calculation of Motif Significance

The Previous Approach. As already mentioned in the introduction, motif detection includes the subtask of determining subgraph significance. In this work, we

$x' = \lceil x \cdot p_d \rceil$ otherwise) and explore exactly these. It can be shown that this does not change the probability of a leaf being explored.

consider the case where the significance of a subgraph is determined by comparing its concentration in the given graph G to its mean concentration $\langle \mathcal{C}_k^i(G) \rangle$ in random graphs with the same degree sequence [15, 9]. It is suggested in [15, 9] to estimate $\langle \mathcal{C}_k^i(G) \rangle$ by generating a large ensemble of random graphs (typically at least 1000) with the same degree sequence as the original graph and then sampling subgraphs in these random graphs. The random graphs are generated from the original graph by randomly switching edges between vertices, which requires a lot of switching operations while at the same time it is never certain when proper randomization has been reached. Also with this method, we are likely to spend lots of excess computational efforts estimating the concentrations of subgraph classes we are not interested in.[5] In this section we propose an algorithm for determining subgraph significance without the need to explicitly generate random graphs (assuming the background model of random graphs with the same degree sequence). We also gain the ability to focus our estimation of significance on specific subgraphs.

Direct Calculation of Subgraph Significance. Milo et al. observe that the total number of size-k subgraphs within an ensemble of large graphs with the same degree sequence does not vary much (see supplementary online material to [14] for details). This allows us to estimate $\langle \mathcal{C}_k^i(G) \rangle$ by

$$\langle \mathcal{C}_k^i(G) \rangle \approx \langle \hat{\mathcal{C}}_k^i(G) \rangle := \frac{\sum_{G' \in \textsc{DegSeq}(G)} |\mathcal{S}_k^i(G')|}{\sum_{G' \in \textsc{DegSeq}(G)} \sum_i |\mathcal{S}_k^i(G')|} \tag{2}$$

where $\textsc{DegSeq}(G)$ is the set of all graphs G' that have the same degree sequence as G. Since all graphs G' can be viewed as graphs over the same set of vertices (because they differ only in their edge sets), Equation (2) can also be written as

$$\langle \hat{\mathcal{C}}_k^i(G) \rangle = \frac{\sum_{\{v_1,\ldots,v_k\} \subseteq V} |\{G' \in \textsc{DegSeq}(G) \mid G'[\{v_1, \ldots, v_k\}] \in \mathcal{S}_k^i\}|}{\sum_{\{v_1,\ldots,v_k\} \subseteq V} |\{G' \in \textsc{DegSeq}(G) \mid G'[\{v_1, \ldots, v_k\}] \text{ connected}\}|} . \tag{3}$$

Both the nominator and denominator of this equation can be estimated in a Monte Carlo approach—i.e., by randomly sampling size-k subsets of the vertices in the input graph—as long as we are able to perform the following calculation: Given G and $\{v_1, \ldots, v_k\}$, find $|\{G' \in \textsc{DegSeq}(G) \mid G'[\{v_1, \ldots, v_k\}] \in \mathcal{S}_k^i\}|$. As it turns out, this number is indeed possible to calculate using two theorems (one for undirected graphs and one for the directed case) due to Bender and Canfield [3, 4]. Without going into technical details here, these theorems allow us to calculate for a given degree sequence how many graphs there are which realize exactly this degree sequence under the constraint that a certain subgraph is fixed. Given a subgraph class $\mathcal{S}_k^i$ and k vertices $\{v_1, \ldots, v_k\}$, we can thus consider all (at most $k!$) ways in which $\{v_1, \ldots, v_k\}$ can induce a subgraph from $\mathcal{S}_k^i$ and hence estimate the nominator in Equation (3).

[5] This is especially important for sparse networks where a randomly sampled subgraph is likely to be a tree. Trees, however, are often considered to be uninteresting motifs [5].

Table 1. Number of size-k subgraphs and the number of respective subgraph classes that occur in our test instances for $3 \leq k \leq 6$. (All instances are directed graphs).

	nodes	edges	subgraphs				subgraph classes			
			size-3	size-4	size-5	size-6	size-3	size-4	size-5	size-6
COLI	423	519	5 206	83 893	1 433 502	22 532 584	4	17	83	390
YEAST	688	1 079	13 150	183 174	2 508 149	32 883 898	7	33	173	888
ELEGANS	306	2 345	47 322	1 394 259	43 256 069	1 309 307 357	13	197	7 071	286 375
YTHAN	135	597	9 487	169 733	2 908 118	45 889 039	8	57	629	9 339

An analogous approach (considering all ways in which the given vertices can be connected) can be used to estimate the denominator in Equation (3). Omitting the details here, it is possible to show that this does not require the explicit consideration of every connected size-k subgraph but only of k^{k-2} subgraphs for undirected graphs and $2 \cdot (2k)^{k-2}$ in the directed case. At first glance it might seem as if this is prohibitively expensive to calculate, but for two reasons it actually promises a gain in efficiency: Firstly, the denominator in Equation (3) is the same for all subgraph classes and hence has to be calculated only once. Secondly, the number of occurring subgraph classes is often far less than the total number of subgraphs (see Table 1). Experiments which are discussed in the next section confirm this expected performance gain.

4 Experimental Studies

Method and Results. We have implemented our algorithms from Sections 2 and 3 in C++. The source code is freely obtainable online at http://www.minet.uni-jena.de/~wernicke/motifs/. As a comparison, we used the *mfinder* 1.1 tool[6] by Kashtan et al. which implements the ESA algorithm. All tests were performed on an AMD Athlon 64 3400+ with 2.4 GHz, 512 KB cache, and 1 GB main memory running under the Debian GNU/Linux 3.1 operating system. Sources were compiled with the GNU gcc/g++ 3.4.3 compiler using the option "-O3."
The network instances for testing the algorithms were up-to-date versions of the motif detection testbed used by Kashtan et al. [9]. The testbed consists of the instances COLI (transcriptional network of *Escherichia Coli* [18]), YEAST (transcriptional network of *Saccharomyces Cerevisiae* [15]), ELEGANS (neuronal network of *Caenorhabditis Elegans* [9]), and YTHAN (food web of the Ythan estuary [21]). Some properties of these networks are summarized in Table 1. The algorithms were compared both for their speed and quality; results and some details as to the experimental setting are shown in Figure 3 and Table 2.

Discussion. Most notable in Figure 3a, RAND-ESU is much faster than the ESA sampling in *mfinder*. This amounts to several orders of magnitude for larger subgraphs ($k \geq 5$). For small sampling quantities, the "coarse" variant of RAND-ESU proved to be faster than the "fine" variant (not explicitly shown in Figure 3a). However, Figure 3b shows that the resulting sampling quality from

[6] Source at http://www.weizmann.ac.il/mcb/UriAlon/groupNetworkMotifSW.html

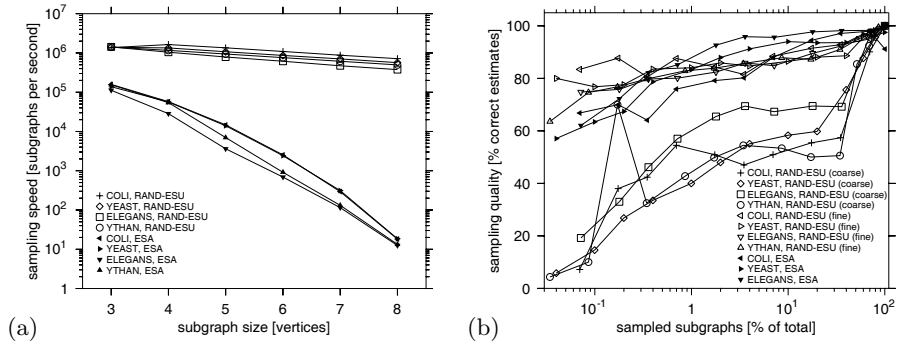

Fig. 3. (a) Sampling speed for different subgraph sizes on a semi-log scale (time measurement does not include the grouping of sampled subgraphs into classes). For ESU, the curve shows the mean speed for three different settings of $(p_1, \ldots, p_k)$ that lead to sampling of an expected 10% of all subgraphs: $(1, \ldots, 1, .316, .316)$, $(1, \ldots, 1, .5, .2)$, and $(1, \ldots, 1, .1)$. (The speed of the deterministic ESU algorithm—not shown here—is slightly faster than that of RAND-ESU.) **(b)** Sampling quality for size-5 subgraphs (size-4 for YTHAN) versus the percentage p of sampled subgraphs (semi-log scale). We define the sampling quality as the percentage of subgraph classes $\mathcal{S}_i^k$ for which $\mathcal{C}_i^k$ is estimated with at most 20% relative error (considering only those subgraph classes for a given p that we would expect to sample at least 10 times on average). RAND-ESU was run with two different settings of the p_d values we refer to as "coarse" $(1, \ldots, 1, \sqrt{p}, \sqrt{p})$ and "fine" $(1, \ldots, 1, p)$. For our YTHAN instance, the *mfinder* tool reproducibly failed to report results for more than 100 samples, hence this curve is not shown.

using "coarse" settings for the p_d values is relatively low when compared to that of ESA. The qualities are roughly equal for the "fine" variant with ESA having a slight advantage for sampling sizes above 1% and close to 100%. (Note that for 100%, RAND-ESU is equivalent to ESU and the results are exact.) Two things are to be noted in this respect, though: Firstly, RAND-ESU is much faster and can, e.g., fully enumerate all size-5 subgraphs in roughly the same time that ESA needs to sample 1% of them. Secondly, the sampling quality of the "fine" variant appears to be more consistent for different networks, e.g., in some percentage ranges ESA has a very good sampling quality for ELEGANS and a comparably fair one for COLI whereas the "fine" RAND-ESU remains much more consistent here. Also note that—contrary to ESA—statistical estimates about the achieved sampling quality can be made with RAND-ESU because of its unbiasedness and the ability to estimate the total number of subgraphs (especially with the "fine" variant where individual samples are fully independent of each other).

As to the estimation of subgraph significance, Table 2 shows that for most subgraphs with $\langle \mathcal{C}_k^i(G) \rangle > 10^{-5}$, $\langle \hat{\mathcal{C}}_k^i(G) \rangle$ is a good approximation in our experimental setting. Further research should investigate the few exceptions, which might hint that for some subgraphs a larger number of samples is needed. Given that direct calculation with our tool was much faster than the explicit generation of random networks, further investigation in this respect appears to be worthwhile. Also, note that with our new approach, the frequency of some sub-

Table 2. For directed size-3 subgraphs, the table shows the approximate subgraph concentrations in random graphs based on the methods discussed in Section 3. For estimating $\langle \mathcal{C}_k^i(G) \rangle$, 10 000 random graphs were generated. The $\langle \hat{\mathcal{C}}_k^i(G) \rangle$ values are based on 100 000 samples. Compared to the $\langle \mathcal{C}_k^i(G) \rangle$ values, their calculation was a few hundred times faster on our machine. For most subgraphs with $\langle \mathcal{C}_k^i(G) \rangle > 10^{-5}$, $\langle \hat{\mathcal{C}}_k^i(G) \rangle$ appears to be a good approximation for $\langle \mathcal{C}_k^i(G) \rangle$ (since often here, the ratio $\langle \mathcal{C}_k^i(G) \rangle / \langle \hat{\mathcal{C}}_k^i(G) \rangle$ is close to one).

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

graphs could be estimated for which the explicit generation of subgraphs did not give any results due to an extremely low average concentration in the explicitly generated random graphs.

5 Conclusion

Based on a detailed analysis of previous approaches we have presented new algorithmic techniques which allow for a faster detection of network motifs and offer useful additional features such as unbiased subgraph sampling and a specifically targeted detection of subgraph significance. This enables motif detection for larger motifs and larger networks than was previously possible and hopefully facilitates future research in the field.

Further research could improve the presented sampling technique, e.g., by examining how the labeling of the vertices in the input graph affects the sampling quality or seeing if RAND-ESU can be tweaked to selectively sample "interesting" parts of the input graph. For subgraph significance, we have shown that a direct calculation scheme may serve as a fast and accurate alternative to the explicit generation of random networks. It would be interesting to further explore this path by extending the scheme to classes of random background models other than those that solely preserve the degree sequence.

Acknowledgments. The author is grateful to Jens Gramm (Tübingen), Falk Hüffner (Jena), and Rolf Niedermeier (Jena) for helpful discussions and comments and to an anonymous referee of WABI 2005 for some insightful remarks on this work.

References

1. I. Albert and R. Albert. Conserved network motifs allow protein-protein interaction prediction. *Bioinformatics*, 20(18):3346–3352, 2004.
2. 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 designed and evolved networks". *Science*, 305:1007c, 2004.
3. E. A. Bender. The asymptotic number of non-negative matrices with given row and column sums. *Disc. Appl. Math.*, 10:217–223, 1974.
4. E. A. Bender and E. R. Canfield. The asymptotic number of labeled graphs with given degree sequences. *J. Comb. Theor. A*, 24:296–307, 1978.
5. J. Berg and M. Lässig. Local graph alignment and motif search in biological networks. *PNAS*, 101(41):14689–14694, 2004.
6. R. A. Duke, H. Lefmann, and V. Rödl. A fast approximation algorithm for computing the frequencies of subgraphs in a given graph. *SIAM J. Comp.*, 24(3):598–620, 1995.
7. S. Itzkovitz, R. Levitt, N. Kashtan, et al. Coarse-graining and self-dissimilarity of complex networks. *Phys. Rev. E*, 71:016127, 2005.
8. S. Itzkovitz, R. Milo, N. Kashtan, et al. Subgraphs in random networks. *Phys. Rev. E*, 68(026127), 2003.
9. 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.
10. D. E. Knuth. Estimating the efficiency of backtrack programs. In *Selected papers on Analysis of Algorithms*. Stanford Junior University, Palo Alto, 2000.
11. T. I. Lee, N. J. Rinaldi, F. Robert, et al. Transcriptional regulatory networks in Saccharomyces Cerevisiae. *Science*, 298:799–804, 2002.
12. B. D. McKay. Practical graph isomorphism. *Congr. Numer.*, 30:45–87, 1981.
13. R. Milo, S. Itzkovitz, N. Kashtan, et al. Response to comment on "network motifs: Simple building blocks of complex networks" and "superfamilies of designed and evolved networks". *Science*, 305:1007d, 2004.
14. R. Milo, S. Itzkovitz, N. Kashtan, et al. Superfamilies of designed and evolved networks. *Science*, 303(5663):1538–1542, 2004.
15. R. Milo, S. S. Shen-Orr, S. Itzkovitz, et al. Network motifs: Simple building blocks of complex networks. *Science*, 298(5594):824–827, 2002.
16. M. E. J. Newman, S. H. Strogatz, and D. J. Watts. Random graphs with arbitrary degree distributions and their applications. *Phys. Rev. E*, 64:026118, 2001.
17. S. Ott, A. Hansen, S. Kim, and S. Miyano. Superiority of network motifs over optimal networks and an application to the revelation of gene network evolution. *Bioinformatics*, 21(2):227–238, 2005.
18. S. S. Shen-Orr, R. Milo, S. Mangan, and U. Alon. Network motifs in the transcriptional regulation network of Escherichia Coli. *Nature Gen.*, 31(1):64–68, 2002.
19. A. Vázquez, R. Dobrin, D. Sergi, et al. The topological relationship between the large-scale attributes and local interaction patterns of complex networks. *PNAS*, 101(52):17940–17945, 2004.
20. A. Vespignani. Evolution thinks modular. *Nature Gen.*, 35(2):118–119, 2003.
21. R. J. Williams and N. D. Martinez. Simple rules yield complex food webs. *Nature*, 404:180–183, 2000.