Approximating the number of Network Motifs

Mira Gonen[⋆] and Yuval Shavitt[⋆⋆]

Tel-Aviv University, Ramat Aviv, Israel

Abstract. World Wide Web, the Internet, coupled biological and chemical systems, neural networks, and social interacting species, are only a few examples of systems composed by a large number of highly interconnected dynamical units. These networks contain characteristic patterns, termed *network motifs*, which occur far more often than in randomized networks with the same degree sequence. Several algorithms have been suggested for counting or detecting the number of induced or non-induced occurrences of network motifs in the form of trees and bounded treewidth subgraphs of size $O(\log n)$, and of size at most 7 for some motifs.
In addition, counting the number of motifs a node is part of was recently suggested as a method to classify nodes in the network. The promise is that the distribution of motifs a node participate in is an indication of its function in the network. Therefore, counting the number of network motifs *a node is part of* provides a major challenge. However, no such practical algorithm exists.
We present several algorithms with time complexity $O\left(e^{2k}k \cdot n \cdot |E| \cdot \log \frac{1}{\delta}/\epsilon^2\right)$ that, for the first time, approximate for every vertex the number of non-induced occurrences of the motif the vertex is part of, for k-length cycles, k-length cycles with a chord, and $(k-1)$-length paths, where $k = O(\log n)$, and for all motifs of size of at most four. In addition, we show algorithms that approximate the total number of non-induced occurrences of these network motifs, when no efficient algorithm exists. Some of our algorithms use the color coding technique.

1 Introduction

1.1 Background and Motivation

World Wide Web, the Internet, coupled biological and chemical systems, neural networks, and social interacting species, are only a few examples of systems composed by a large number of highly interconnected dynamical units. The first approach to capture the global properties of such systems is to model them as graphs whose nodes represent the dynamical units, and whose links stand for the interactions between them. Such networks have been extensively studied by exploring their global topological features such as power-law degree distribution, the existence of dense-core and small diameter. (For references see the full version of the paper [7]). However, two networks which have similar global features

[⋆] Email: `gonenmir@post.tau.ac.il`.

[⋆⋆] Email: `shavitt@eng.tau.ac.il`.

can have significant differences in structure, which can be captured by examining local structures they include: e.g., one of them may include a specific subgraph many more times than the other. Therefore these small subgraphs, termed network motifs, were suggested to be elementary building blocks that carry out key functions in the network. Milo et al. [12] found motifs in networks from biochemistry, neurobiology, ecology, and engineering. Specifically, they found motifs in the World Wide Web. Moreover, Hales and Arteconi [9] presented results from a motif analysis of networks produced by peer-to-peer protocols. They showed that the motif profiles of such networks closely match protein structure networks. Thus efficiently detecting and counting the number of network motifs provide a major challenge.

As a result novel computational tools have been developed for *counting* subgraphs in a network and *discovering* network motifs. Most of existing work deal with *induced* motifs, while there is also work that focuses on *non-induced* motifs.[1] The motivation for considering non-induced subgraphs is that the process of obtaining large networks (such as the AS graph) are far from complete and error free; they lack existing edges. Thus, an occurrence of a specific network motif in one network may include additional edges in its occurrence in another network and vice versa.

Several existing algorithms for counting and detection non-induced motifs [6, 4, 2, 1, 16] used the color coding technique of Alon et al. [3]. Color coding is an innovative combinatorial approach that was introduced by Alon et al. [3] to detect simple paths, trees and bounded treewidth subgraphs in unlabeled graphs. Color coding is based on assigning random colors to the vertices of an input graph. It considers only those subgraphs where each vertex has a unique color. Such colorful subgraphs can then be detected through efficient use of dynamic programming, in time polynomial with n, the size of the input graph. If the above procedure is repeated sufficiently many times (polynomial with n, provided that the subgraph we are looking for is of size $O(\log n)$, it is guaranteed that a specific occurrence of the query subgraph will be detected with high probability. The color coding technique is a building block in some of the algorithms presented in this paper.

Przulj et al. [14] described how to *count* all *induced* subgraphs with up to 5 vertices in a PPI (Protein-Protein Interaction) network. Faster techniques that count induced subgraphs of size up to 6 were developed by Hormozdiari et al. [10], and for size up to 7 were shown by Grochow and Kellis [8]. The running time of these techniques all increase exponentially with the size of the motif. Kashtan et al. [11] showed an algorithm for *detecting* induced network motifs that sample the network. This algorithm detect induced occurrences of small motifs (motifs with $k \leq 7$ vertices). Wernicke et al. [17] claims that Kashtan et al.'s algorithm suffers from a sampling bias and scales poorly with increasing subgraph size. Thus, Wernicke [17] presented an improved algorithm for network

[1] G_0 is an induced subgraph of a graph G if and only if for each pair of vertices v_0 and w_0 in G_0 and their corresponding vertices v and w in G there is an edge between v_0 and w_0 in G_0 if and only if there is an edge between v and w in G.

motif detection which overcomes these drawbacks. Scott et al. [15] focused on the subgraph detection problem. Dost et al. [6] showed how to solve the subgraph detection problem for subgraphs of size $O(\log n)$, provided that the query subgraph is either a simple path, a tree, or a bounded treewidth subgraph. Arvind and Raman [4] *counted* the number of subgraphs in a given graph G which are isomorphic to a bounded treewidth graph H. They gave a randomized approximate counting algorithm with a running time of $k^{O(k)} \cdot n^{b+O(1)}$, where n and k are the number of vertices in G and H, respectively, and b is the treewidth of H. Alon and Gutner [2] combined the color coding technique with a construction of Balanced Families of Perfect Hash Functions to obtain a deterministic algorithm to count the number of simple paths or cycles of size k in an input graph G. Alon et al. [1] improved the algorithm of Alon and Gutner. They presented a polynomial time algorithm for approximating the number of non-induced occurrences of trees and bounded treewidth subgraphs with $k = O(\log n)$ vertices with a running time of $2^{O(k \log \log k)} \cdot n^{O(1)}$.

A new systematic measure of a network's *local* topology was recently suggested by Przulj [13]. They term this measure "graphlet distribution" of a vertex. Namely, they count for each vertex the number of all motifs of size at most five that adjacent to the vertex. The promise is that the distribution of motifs a node participate in is an indication of its function in the network, thus nodes can be divided into functional classes. In addition, Becchetti et al [5] have recently shown that the distribution of the local number of triangles and the related clustering coefficient can be used to detect the presence of spamming activity in large scale Web graphs, as well as to provide useful features for the analysis of biochemical networks or the assessment of content quality in social networks. Therefore, counting the number of network motifs a node is part of also provides a major challenge. However, no practical algorithm for counting the number of network motifs a node is part of exists.

1.2 Our Contributions

We present several algorithms with time complexity $O\left(e^{2k} k \cdot n \cdot |E| \cdot \log \frac{1}{\delta} / \epsilon^2\right)$ that, for the first time, approximate for every vertex the number of non-induced occurrences of the motif the vertex is part of, for k-length cycles, k-length cycles with a chord, and $(k-1)$-length paths, where $k = O(\log n)$. We also provide algorithms with time complexity $O\left(n \cdot |E| \cdot \log \frac{1}{\delta} / \epsilon^2 + |E|^2 + |E| \cdot n \log n\right)$ that, for the first time, approximate for every vertex the number of non-induced occurrences of the motif the vertex is part of for all motifs of size of at most four. In addition, we show an $O\left(e^k k \cdot n \cdot |E| \cdot \log \frac{1}{\delta} / \epsilon^2\right)$ algorithm that, for the first time, approximates the total number of non-induced occurrences of $O(\log n)$-length cycles with a chord. Moreover, we improve the time complexity of approximating the total number of non-induced occurrences of "tailed" triangles and 4-cliques upon existing algorithms. Some of our algorithms use the color coding technique of Alon et al. [3].

Organization: In Section 2 we give notations and definitions. In Section 3 we introduce motifs counting approximation algorithms for $O(\log n)$-size motifs.

In Section 4 we present motifs counting algorithms for all four-size motifs. We summarize our conclusions in Section 5.

2 Preliminaries

Let $G = (V, E)$ be an undirected graph with n vertices. We assume that G is represented by an adjacency lists. For a vertex v let $N(v)$ denote the set of neighbors of v and let $\deg(v)$ denote the degree of v. A motif H is said to be isomorphic to a subgraph H' in G if there is a bijection between the vertices of H and the vertices of H' such that for every edge between two vertices v and u of H there is an edge between the vertices v' and u' in H' that correspond to v and u respectively. Such a subgraph H' is considered to be a non-induced occurrence of H in G. For a vertex v we say that v is *adjacent* to H if v is a vertex of H. Denote by $[k]$ the set $\{1, \ldots, k\}$. Denote by $col(v)$ the color of vertex v.

Let H be a motif with k vertices, and let $G = (V, E)$ be a graph where $|V| = n$. Assign a color to each vertex of V from the color set $[k]$. The colors are assigned to each vertex independently and uniformly at random. A copy of H in G is said to be *colorful* if each vertex on it is colored by a distinct color.

For a problem f, let $\#f$ denote the number of distinct solutions of f.

Definition 1. ***((ϵ, δ)-approximation)*** *An algorithm $\mathcal{A}$ for a counting problem f is a (ϵ, δ)-approximation if it takes an input instance and two real values ϵ, δ and produces an output y such that*

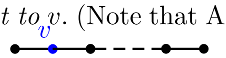
$$\Pr[(1 - \epsilon) \cdot \#f \leq y \leq (1 + \epsilon) \cdot \#f] \geq 1 - 2\delta.$$

3 Algorithms for Counting Motifs of size $O(\log n)$

Given a graph $G = (V, E)$ and a vertex v, we describe how to approximately count for every vertex v the number of non-induced occurrences of $(k-1)$-length paths, k-length cycles, and k-length cycles with a chord that are adjacent to v, for $k = O(\log n)$. In addition, for each such motif H we present an algorithm for approximating the number of non-induced subgraphs of G that are isomorphic to H when no efficient algorithm exists. Most of our approximation algorithms apply the color coding technique of Alon et al. [3]. Note that we allow overlaps between the motifs we count, i.e. two occurrences of H, namely H' and H'' may share vertices; in fact the vertex sets of H' and H'' may be identical. We consider H' and H'' distinct occurrences of H provided that the edge sets of H' and H'' are not identical.

3.1 Counting Paths

In this section assume that H is a simple path of length $k-1$.
We present an algorithm to approximately count for every vertex v, the number

of subgraphs of G which are isomorphic to H and *adjacent to* v. (Note that Alon et al [1] only count the total number of paths in the graph). 

Let $t = \log(1/\delta)$, and let $s = \frac{4k^k}{\epsilon^2 k!}$. Assume that we have a k-coloring of G, i.e., each vertex is randomly and independently colored with a color in $[k]$. For each vertex v and each subset S of the color set $[k]$, let $P_i(v, S)$ be the number of colorful paths adjacent to v using colors in S at the ith coloring, and let $C_i(v, S)$ be the number of colorful paths for which one of their endpoint is v using colors in S at the ith coloring.

Consider the following algorithm. The algorithm takes as input: a graph $G = (V, E)$, a vertex $v \in V$, the requested path length $k - 1$, fault-tolerance ϵ, and an error probability δ.

Algorithm 1 (A (ϵ, δ)-approximation algorithm for counting simple paths of length $k - 1$ adjacent to a vertex v)

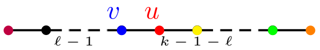
```
1. For $j = 1$ to $t$
   (a) For $i = 1$ to $s$
       i. Color each vertex of $G$ independently and uniformly at random with
          one of the $k$ colors.
       ii. For all $u \in V$ $C_i(u, \phi) = 1$.
       iii. For all $\ell \in [k]$ $C_i(v, \{\ell\}) = \begin{cases} 1 & \textit{if } col(v) = \ell; \\ 0 & \textit{otherwise.} \end{cases}$
       iv. For all $S \subseteq [k]$ s.t $|S| > 1$ $C_i(v, S) = \sum_{u \in N(v)} C_i(u, S \setminus \{col(v)\})$.
       v. $P_i(v, [k]) = \sum_{\ell=1}^{k} \sum_{u \in N(v)} \sum_{(S_1, S_2) \in A_{\ell,v}} C_i(v, S_1) \cdot C_i(u, S_2)$,
          where $A_{\ell,v} = \{(S_i, S_j) | S_i \subseteq [k], S_j \subseteq [k], S_i \cap S_j = \phi, |S_i| = \ell, |S_j| = k - \ell\}$.
       vi. Let $X_i^v = P_i(v, [k])$.
   (b) Let $Y_j^v = \frac{\sum_{i=1}^{s} X_i^v}{s}$.
2. Let $Z^v$ be the median of $Y_1^v, ..., Y_t^v$.
3. Return $Z^v \cdot k^k/k!$.
```

Our main theorem is the following:

Theorem 1 *Let $G = (V, E)$ be an undirected graph, and let H be a simple path of length $k - 1$. Then for all $v \in V$ Algorithm 1 is a (ϵ, δ)-approximation for the number of copies of H in G that are adjacent to v, with time complexity $O\left(e^k |E| \log(1/\delta)/\epsilon^2\right)$.*

For proving Theorem 1 we first prove the following lemma:

Lemma 1 *For all $v \in V$ $P_i(v, [k])$ can be computed in $O(2^k |E|)$ time.*

Proof: According to Alon et al [1] the time complexity for computing $C_i(v, S)$ for all v and S is $\sum_{v \in V} \sum_{u \in N(v)} O(2^k) + \sum_{v \in V} O(k) = O(\sum_{v \in V} \deg(v) \cdot 2^k) = O(2^k(|E|))$. A vertex v is adjacent to a colorful path of length $k - 1$ if and only if it is an endpoint of a colorful path of length $\ell - 1$, one of its neighbors is an endpoint of a colorful path of length $k - 1 - \ell$, and the subsets of colors of both paths are disjoint. 

Thus, for each vertex v, the number of colorful paths of length $k-1$ that are adjacent to v is

$$P(v,[k]) = \sum_{\ell=1}^{k} \sum_{u \in N(v)} \sum_{(S_1,S_2)\in A_{\ell,v}} C(v,S_1)\cdot C(u,S_2),$$

where $A_{\ell,v} = \{(S_i,S_j) | S_i \subseteq [k], S_j \subseteq [k], S_i \cap S_j = \phi, |S_i| = \ell, |S_j| = k-\ell\}$. (We define $C(u,\phi)=1$ for every vertex u). Therefore the running time for computing $P(v,[k])$ for all v, assuming that $C(u,S)$ is known for any vertex u and any color set S, is:

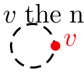
$$\sum_{v\in V}\sum_{\ell=1}^{k}\binom{k}{\ell}\deg(v) = 2^{k-1}\cdot 2\cdot |E|.$$

Thus the total running time for computing $P(v,[k])$ for all v is $O(2^k|E|)$. □

The proof of Theorem 1 is based on Lemma 1 and the approximation technique of Alon et al. [1]. The details of the proof of Theorem 1 appear in the full version of the paper [7].

3.2 Counting Cycles

In this section assume that H is a simple cycle of length k.

We present an algorithm to approximately count for every vertex v the number of subgraphs of G which are isomorphic to H and adjacent to v. 

Let $t = \log(1/\delta)$, and let $s = \frac{4k^k}{\epsilon^2 k!}$. Assume that we have a k-coloring of G, i.e., each vertex is randomly and independently colored with a color in $[k]$. For each pair of vertices v, x and each color set S of the color set $[k]$ let $C_i(v,x,S)$ be the number of colorful paths between v and x using colors in S at the ith coloring, and let $CY_i(v,S)$ be the number of colorful cycles adjacent to v using colors in S at the ith coloring.

Consider the following algorithm. The algorithm takes as input: a graph $G=(V,E)$, a vertex $v\in V$, the requested cycle length k, fault-tolerance ϵ, and an error probability δ. The algorithm uses a procedure to compute the number of colorful paths between v and any other vertex.

Algorithm 2 (A (ϵ,δ)-approximation algorithm for counting simple cycles of length k adjacent to a vertex v)

```
1. For $j = 1$ to $t$
   (a) For $i = 1$ to $s$
         i. Color each vertex of $G$ independently and uniformly at random with
            one of the $k$ colors.
        ii. For all $x \in V$ $C_i(v,x,[k]) = count\text{-}path(v,x,k)$.
       iii. Let $CY_i(v,[k]) = \frac{1}{2}\sum_{u\in N(v)} C_i(v,u,[k])$.
        iv. Let $X_i^v = CY_i(v,[k])$.
   (b) Let $Y_j^v = \frac{\sum_{i=1}^{s} X_i^v}{s}$.
```

2. *Let Z^v be the median of $Y_1^v, ..., Y_t^v$.*
3. *Return $Z^v \cdot k^k/k!$.*

Algorithm 3 count-path(v,x,k)(counting simple paths of length $k-1$ between v and x)

1. *For all $S \subseteq [k]$ s.t $S = \{\ell\}$*

$$C_i(v, x, S) = \begin{cases} 1 \textit{ if } col_i(v) = col_i(x) = \ell; \\ 0 \textit{ otherwise.} \end{cases}$$

2. *For $q = 2$ to k, for all $S \subseteq [k]$ s.t $|S| = q$*

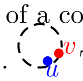
$$C_i(v, x, S) = \sum_{u \in N(v)} C_i(u, x, S \setminus \{col_i(v)\}).$$

Our main theorem is the following:

Theorem 2 *Let $G = (V, E)$ be an undirected graph, and let H be a simple cycle of length k. Then for every vertex v Algorithm 2 is a (ϵ, δ)-approximation for the number of copies of H in G that are adjacent to v, with time complexity $O\left(e^k \cdot k \cdot n \cdot |E| \log(1/\delta)/\epsilon^2\right)$.*

For proving Theorem 2 we first prove the following lemma:

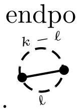
Lemma 2 *For all $v \in V$ $CY_i(v, [k])$ can be computed in $O(2^k \cdot k \cdot n \cdot |E|)$ time.*

Proof: A vertex v is adjacent to a colorful cycle of length k if and only if it is an endpoint of a colorful path of length $k-1$, which has one of v's neighbor as an endpoint. 

Therefore, we first compute for every two vertices u, v the number of colorful paths of length $k-1$ between u and v. If u is a neighbor of v, then we get a cycle of length k. The running time for computing $CY(v, [k])$ for all v is then $2^k \left(\sum_{v \in V} 1 + \sum_{v,x \in V} 1 + (k-2) \sum_{v,x \in V} \deg(v)\right) = O(2^k \cdot k \cdot n \cdot |E|)$. □

Proof of Theorem 2. The correctness of the approximation returned by Algorithm 2 is proved in the same manner as in the proof of Theorem 1. Lemma 2 implies the correctness of the computation of $CY_i(v, [k])$. The time complexity of Algorithm 2 is $O\left(e^k \cdot k \cdot n \cdot |E| \log(1/\delta)/\epsilon^2\right)$ by Lemma 2, and by showing that the number of colorings used by the algorithm is $O\left(e^k \log(1/\delta)/\epsilon^2\right)$. (This is proved in the same manner as in the proof of Theorem 1). This completes the proof. □

3.3 Counting k-length Cycles with a chord

In this section assume that H is a simple cycle of length k with a chord, such that the distance between the endpoint of the chord on the cycle is $\min\{\ell, k-\ell\}$, for some given $2 \le \ell \le k-2$. 

We present an algorithm to approximately compute the number of subgraphs of G which are isomorphic to H, and, for every vertex v, the number of subgraphs of G which are isomorphic to H and adjacent to v.

The approximation of the number of colorful subgraphs of G which are isomorphic to H appears in the full version of the paper [7].

We now approximate for every $v \in V$ the number of colorful subgraphs of G which are isomorphic to H and are adjacent to v. Let $t = \log(1/\delta)$, let $s = \frac{4 \cdot k^k}{\epsilon^2 k!}$. Assume that we have a k-coloring of G, i.e., each vertex is randomly and independently colored with a color in $[k]$. Let $P_i(v, u, w, S)$ be the number of colorful paths from u to w that are adjacent to v in the ith coloring, using the colors in S. Recall that $C_i(v, u, S)$ is the number of colorful paths from v to u in the ith coloring, using the colors in S. Let $A_{V'}^{z,b}(S) = \{(S_1, S_2) | S_1, S_2 \subseteq [k], |S_1| = z+1, |S_2| = b-z+1, S_1 \cup S_2 = S, S_1 \setminus \{col(u) | u \in V'\} \cap S_2 \setminus \{col(u) | u \in V'\} = \phi\}$. Consider the following algorithm. The algorithm takes as input: a graph $G = (V, E)$, a vertex v, fault-tolerance ϵ, and an error probability δ.

Algorithm 4 (A (ϵ, δ)-approximation algorithm for counting simple cycles of length k with a chord that are adjacent to v)

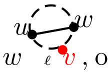
```
1. For $j = 1$ to $t$
   (a) For $i = 1$ to $s$
       i. Color each vertex of $G$ independently and uniformly at random with
          one of the $k$ colors.
       ii. $X_i^v = 0$.
       iii. For every edge $(u, w) \in E$ :
       iv. For all $S \subseteq [k]$ s.t $|S| = \ell + 1$
           $P_i(v, u, w, S) = \sum_{z=1}^{\ell-1} \sum_{(S_1, S_2) \in A_v^{z,\ell}(S)} C_i(v, w, S_1) \cdot C_i(v, u, S_2)$.
       v. Let $X_i^v = X_i^v + \sum_{(S_3, S_4) \in A_{uw}^{\ell,k}([k])} P_i(v, u, w, S_3) \cdot C_i(u, w, S_4) +$
          $\sum_{(S_3, S_4) \in A_{uw}^{k-\ell,k}([k])} P_i(v, u, w, S_3) \cdot C_i(u, w, S_4) +$
          $\sum_{(S_3, S_4) \in A_{uv}^{\ell,k}([k])} C_i(v, u, S_3) \cdot C_i(v, u, S_4)$.
   (b) Let $Y_j^v = \frac{\sum_{i=1}^{s} X_i^v}{s}$.
2. Let $Z^v$ be the median of $Y_1^v, ..., Y_t^v$.
3. Return $Z^v \cdot k^k / k!$.
```

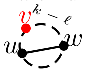
Our main theorem is the following:

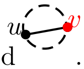
Theorem 3 *Let $G = (V, E)$ be an undirected graph, and let H be a simple cycle of length k with a chord. Then, for every $v \in V$, Algorithm 4 is a (ϵ, δ)-approximation for the number of copies of H in G that are adjacent to v, with time complexity $O\left(|E| \cdot n \cdot e^{2k} \cdot k \log(1/\delta)/\epsilon^2\right)$.*

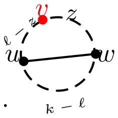
For proving Theorem 3 we first prove the following lemma:

Lemma 3 *X_i^v can be computed with time complexity $O(|E| \cdot n \cdot 2^{2k} \cdot k)$.*

Proof: Let (u, w) be the chord, and let ℓ be the distance on the cycle between u and w. The number of copies of H that are adjacent to v depends on the position of v. There are three cases: one for which v is adjacent to a path of length ℓ between u and w 

, one for which v is adjacent to a path of length $k-\ell$ between u and w 

, and one for which v is an endpoint of the chord 

. In the first case we first count all the colorful paths of length z between v and w and multiply it by the number of colorful paths of length $\ell - z$ between v and u, where $1 \leq z \leq \ell - 1$. (W.l.o.g assume that $\ell \neq k - \ell$). 

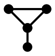
Thus for all $S \subseteq [k]$ s.t $|S| = \ell + 1$ $P_i(v, u, w, S) = \sum_{z=1}^{\ell-1} \sum_{(S_1,S_2) \in A_v^{z,\ell}(S)} C_i(v, w, S_1) \cdot C_i(v, u, S_2)$. Therefore the total number of copies of H that are adjacent to v in the first case is the number of ℓ-length colorful paths between u and w that are adjacent to v, multiplied by the number of $k-\ell$-length colorful paths between u and w, with disjoint set of colors (except for the colors of u and w): $\sum_{(S_3,S_4) \in A_{uw}^{\ell,k}([k])} P_i(v, u, w, S_3) \cdot C_i(u, w, S_4)$. The second case is computed in the same manner. The third case is computed as follows. We count the number of ℓ-length colorful paths between u and v and multiply it by the number of $k - \ell$-length colorful paths between u and v, using disjoint set of colors besides the colors of u and v. Computing the running time: According to the proof of Lemma 2, the time complexity for computing $C_i(v, w, S)$ for every color-set S and every pair of vertices v, w is $O(2^k \cdot k \cdot n \cdot |E|)$. The running time of computing $P_i(v, u, w, S)$ for all vertices v, u, w, and every color-set S of size $\ell+1$ is $O\left(\sum_{z=1}^{\ell-1} \binom{\ell}{z} \cdot \binom{k}{\ell+1} \cdot |E| \cdot n\right) = O(2^\ell \cdot \binom{k}{\ell} \cdot |E| \cdot n)$. Therefore the time complexity of computing the first case is $O(\sum_{v \in V} \sum_{(u,w) \in E} \binom{k}{\ell+1} + 2^\ell \cdot \binom{k}{\ell} \cdot |E| \cdot n) + O(2^k \cdot k \cdot n \cdot |E|) = O(\binom{k}{\ell} \cdot |E| \cdot n + 2^\ell \cdot \binom{k}{\ell} \cdot |E| \cdot n) + O(2^k \cdot k \cdot n \cdot |E|) = O(2^k \cdot k \cdot n \cdot |E| \cdot 2^\ell)$. In the same manner the time complexity of case two is $O(2^k \cdot k \cdot n \cdot |E| \cdot 2^{k-\ell})$. The time complexity of the third case (besides computing $C_i(v, w, S)$ is $O\left(\sum_{(u,v) \in E} \binom{k}{\ell+1}\right) = O(|E| \cdot \binom{k}{\ell})$. Thus the total time complexity is $O(|E| \cdot n \cdot 2^{2k} \cdot k)$. □

Proof of Theorem 3. The correctness of the approximation returned by Algorithm 4 is proved in the same manner as in the proof of Theorem 1. Lemma 3 implies the correctness of the computation of X_i^v. The time complexity of Algorithm 4 is $O\left(|E| \cdot n \cdot k \cdot e^{2k} \log(1/\delta)/\epsilon^2\right)$ by Lemma 3 and by showing that the number of colorings used by the algorithm is $O\left(e^k \log(1/\delta)/\epsilon^2\right)$. (This is proved in the same manner as in the proof of Theorem 1). This completes the proof. □

4 Algorithms for Counting all four-size Motifs

Given a graph $G = (V, E)$ and a vertex v, we describe how to approximately count for every vertex v the number of non-induced occurrences of each possible motif H that are adjacent to v. In addition, for each such motif H we present an algorithm for approximating the number of non-induced subgraphs of G that are isomorphic to H when no efficient algorithm exists. Note that we allow overlaps between the motifs, as in the previous section.

4.1 Counting "Tailed Triangles"

In this section assume that H is a triangle with a "tail" of length one.  We present an algorithm that approximates the number of subgraphs of G which are isomorphic to H, and, for every vertex v, approximates the number of subgraphs of G which are isomorphic to H and adjacent to v.

We first approximate for every $v \in V$ the number of subgraphs of G which are isomorphic to H and are adjacent to v. Let $TR_G(v)$ be the approximation of the total number of triangles in G that are adjacent to v, according to Algorithm 2. Let $G_v = (V_v, E_v)$, where $V_v = V \setminus \{v\}$, and E_v is the induced set of edges received by removing all edges adjacent to v. Consider the following algorithm. The algorithm takes as input: a graph $G = (V, E)$, a vertex v, fault-tolerance ϵ, and an error probability δ.

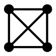
Algorithm 5 (A (ϵ, δ)-approximation algorithm for counting simple "tailed triangles" adjacent to v)

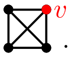
```
1. $TL_G(v) = 0$
2. $TR_G(v)$ = result of Algorithm 2 ($G$,$v$, $k = 3$, $\epsilon$, $\delta$).
3. $TL_G(v) = TL_G(v) + TR_G(v) \cdot (|N(v)| - 2)$.
4. Compute $G_v$ by going over the whole adjacency list and removing $v$ any time
   is appears in the list.
5. For all $u \in N(v)$ $TR_{G_v}(u)$= result of Algorithm 2 ($G_v$,$u$, $k = 3$, $\epsilon$, $\delta$).
6. $TL_G(v) = TL_G(v) + \sum_{u \in N(v)} TR_{G_v}(u)$.
7. $a_v = 0$.
8. For all $u \in N(v)$ go over $v$'s adjacency list, and for each vertex $w$ in $u$'s
   adjacency list check if $w \in N(v)$ by going over $v$'s adjacency list. If $w \in N(v)$
   then $a_v = a_v + \deg w - 2 + \deg u - 2$.
9. Return $TL_G(v) + a_v$.
```

Theorem 4 *Let $G = (V, E)$ be an undirected graph, and let H be a triangle with a "tail" of length one. Then, for every vertex v, the number of copies of G that are isomorphic to H and adjacent to v can be (ϵ, δ)-approximated, with time complexity $O\left(|E|^2 + n \cdot |E| \log(1/\delta)/\epsilon^2\right)$.*

The proof of Theorem 4 and counting the number of subgraphs of G, which are isomorphic to H, appear in the full version of the paper [7].

4.2 Counting 4-Cliques

In this section assume that H is a clique of size four.  We present an algorithm that *exactly* computes the number of subgraphs of G which are isomorphic to H, and, for every vertex v, the number of subgraphs of G which are isomorphic to H and adjacent to v.

We first compute, for every vertex v, the number of subgraphs of G which are isomorphic to H and adjacent to v. 

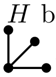
. We run the following algorithm: Let $Cl(v)$ be the number of four-cliques in the graph that are adjacent to v. The algorithm takes as input: a graph $G = (V, E)$, a vertex v.

Algorithm 6 (Algorithm for counting 4-cliques that are adjacent to v)

```
1. $Cl(v) = 0$.
2. For every vertex $u \in N(v)$:
   (a) Compute $N(v) \cap N(u)$:
       i. Go over all the vertices in the adjacency list of $v$ and the adjacency list of $u$, and add each vertex to a list. (Thus a vertex can appear several times in the list).
       ii. Sort the vertices in the list (which is a multiset) according the names of the vertices.
       iii. For each vertex in the list count the number of times it appears in the list. If it appears twice then add the vertex to a list $\ell(u, v)$.
       iv. Sort the list $\ell(u, v)$ according to the names of the vertices.
   (b) For all $w \in \ell(u, v)$ go over the adjacency list of $w$ and for each vertex $t \neq v, u$ in this adjacency list check if $t \in \ell(u, v)$. If $t \in \ell(u, v)$ then $Cl(v) := Cl(v) + 1$.
3. Return $Cl(v)/6$.
```

Theorem 5 *Let $G = (V, E)$ be an undirected graph, and let H be a clique of size four. Then for all $v \in V$ Algorithm 6 counts the number of copies of H in G that are adjacent to v, with time complexity $O(|E| \cdot n \log n + |E|^2)$.*

The proof of Theorem 5 and counting the number of subgraphs of G which are isomorphic to H appear in the full version of the paper [7].

Counting Small Trees Let H be a tree of size four that is consisted of a vertex and three of its neighbors.  Computing the number of subgraphs of G which are isomorphic to H, and, for every vertex v, the number of subgraphs of G which are isomorphic to H and adjacent to v appear in the full version of the paper [7].

5 Conclusions

We presented algorithms with time complexity $O\left(e^{2k}k \cdot n \cdot |E| \cdot \log\frac{1}{\delta}/\epsilon^2\right)$ that, for the first time, approximate the number of non-induced occurrences of the motif a vertex is part of, for k-length cycles, k-length cycles with a chord, and

$(k-1)$-length paths, where $k = O(\log n)$, and for all motifs of size of at most four. In addition, we showed algorithms that approximate the total number of non-induced occurrences of these network motifs, when no efficient algorithm exists. Approximating the number of non-induced occurrences of the motif a vertex is part of, for other motifs of size $O(\log n)$ is left for future work.

Acknowledgment: We thank Dana Ron for many hours of fruitful discussions.

References

1. N. Alon, P. Dao, I. Hajirasouliha, F. Hormozdiari, and S. C. Sahinalp. Biomolecular network motif counting and discovery by color coding. *Bioinformatics*, 1:1–9, 2008.
2. N. Alon and S. Gutner. Balanced families of perfect hash functions and their applications. In *ICALP*, pages 435–446, 2007.
3. N. Alon, R. Yuster, and U. Zwick. Color-coding. *Journal of the ACM*, 42(4):844, 1995.
4. V. Arvind and V. Raman. Approximation algorithms for some parameterized counting problems. In *ISAAC*, pages 453–464, 2002.
5. L. Becchetti, P. Boldi, C. Castillo, and A. Gionis. Efficient semi-streaming algorithms for local triangle counting in massive graphs. In *Proceeding of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 16–24, 2008.
6. B. Dost, T. Shlomi, N. Gupta, E. Ruppin, V. Bafna, and R. Sharan. QNet: A tool for querying protein interaction networks. In *RECOMB*, pages 1–15, 2007.
7. M. Gonen and Y. Shavitt. Approximating the number of network motifs. Technical report, School of Electrical Enjeneering, Tel Aviv University, 2008.
8. J. Grochow and M. Kellis. Network motif discovery using subgraph enumeration and symmetry-breaking. In *RECOMB*, pages 92–106, 2007.
9. D. Hales and S. Arteconi. Motifs in evolving cooperative networks look like protein structure networks. *Special Issue of ECCS'07 in The Journal of Networks and Heterogeneous Media*, 3(2):239–249, 2008.
10. F. Hormozdiari, P. Berenbrink, N. Przulj, and S.C. Sahinalp. Not all scale-free networks are born equal: The role of the seed graph in ppi network evolution. *PLoS: Computational Biology*, 3(7):e118, 2007.
11. 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.
12. R. Milo, S. Shen-Orr, S. Itzkovitz, N. Kashtan, D. Chklovskii, and U. Alon. Network motifs: Simple building blocks of complex networks. *Science*, 298:824–827, 2002.
13. N. Przulj. Biological network comparison using graphlet degree distribution. *Bioinformatics*, 23(2):e177–e183, 2007.
14. N. Przulj, D. G. Corneil, and I. Jurisica. Modeling interactome: Scale- free or geometric? *Bioinformatics*, 150:216–231, 2005.
15. J. Scott, T. Ideker, R. Karp, and R. Sharan. Efficient algorithms for detecting signaling pathways in protein interaction networks. In *RECOMB*, pages 1–13, 2005.
16. T. Shlomi, D. Segal, and E. Ruppin. QPath: a method for querying pathways in a protein-protein interaction network. *Bioinformatics*, 7:199, 2006.
17. S. Wernicke. Efficient detection of network motifs. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 3(4):347–359, October 2006.