# Extending the Applicability of Graphlets to Directed Networks

David Aparício, Pedro Ribeiro and Fernando Silva

**Abstract**—With recent advances in high-throughput cell biology the amount of cellular biological data has grown drastically. Such data is often modeled as graphs (also called networks) and studying them can lead to new insights into molecule-level organization. A possible way to understand their structure is by analysing the smaller components that constitute them, namely network motifs and graphlets. Graphlets are particularly well suited to compare networks and to assess their level of similarity due to the rich topological information that they offer but are almost always used as small undirected graphs of up to five nodes, thus limiting their applicability in directed networks. However, a large set of interesting biological networks such as metabolic, cell signaling or transcriptional regulatory networks are intrinsically directional, and using metrics that ignore edge direction may gravely hinder information extraction. Our main purpose in this work is to extend the applicability of graphlets to directed networks by considering their edge direction, thus providing a powerful basis for the analysis of directed biological networks. We tested our approach on two network sets, one composed of synthetic graphs and another of real directed biological networks, and verified that they were more accurately grouped using directed graphlets than undirected graphlets. It is also evident that directed graphlets offer substantially more topological information than simple graph metrics such as degree distribution or reciprocity. However, enumerating graphlets in large networks is a computationally demanding task. Our implementation addresses this concern by using a state-of-the-art data structure, the g-trie, which is able to greatly reduce the necessary computation. We compared our tool to other state-of-the art methods and verified that it is the fastest general tool for graphlet counting.

**Index Terms**—Pattern matching, Network topology, Graph Algorithms, Graphs and Networks.

## 1 Introduction

The advent of high-throughput cell biology technologies such as DNA microarrays [1] increased the amount of data pertaining to molecular interactions exponentially. While this recent flood of information has greatly contributed to a more accurate understanding of molecule-level organization, it has also created the need to find ways to filter and model this data so that it is rendered intelligible to the practitioner. In the field of computational biology different types of cellular networks are modeled as graphs with nodes representing specific biological components such as proteins or genes, and the physical, chemical or functional interactions between them modeled as edges.

Inspecting the graph's topological features can yield valuable information about the network. If a network has topological features that are not expected to occur in neither purely random nor purely regular graphs it is considered to be a complex network, and most real-world networks are found to be complex. Singular characteristics commonly associated with complex networks include having a small-world structure [2] or a degree distribution that follows a power-law (scale-free networks) [3]. Brain networks, for instance, have been identified as small-world networks [4], meaning that each node (representing either a brain region in mesoscale conectomes or a neuron in microscale connectomes) is only a few connections away from any other node. Furthermore, the distance of the average path length in the brain has been negatively correlated with a person's IQ [5], which seems to indicate the importance of a small-world organization to networks' efficiency. Other statistics such as the number of connected components and the clustering coefficient are also frequently used to characterize a network.

Another approach to uncover the underlying structure of complex networks is to decompose them into their smaller components or *subgraphs*. Obtaining the frequency of each type of subgraph offers detailed topological information which can be used to summarize and compare networks. Frequent subgraph mining (FSM) algorithms can identify biological pathways prevalent in many species or common fragments shared by distinct molecules [6]. The aim of FSM is to find subgraphs that appear frequently in an ensemble of networks; however, depending on the researcher's goal, it might be more insightful to discover subgraphs that are *overrepresented* just on a single network. Network motifs are small overrepresented subgraphs described by Milo et al. [7] as the building blocks of large complex networks. Studies such as the one by Prill et al. [8] support this view for biological networks and network motifs have since been used to analyze a wide range of cellular biological networks such as protein-protein interaction networks [9], transcriptional regulatory networks [10], metabolism networks [11] or cell signaling networks [12].

It is often useful to compare networks against each other, particularly because if a given network's properties are known it allows for knowledge transfer based on the similarity or difference between two networks. One way to perform this comparison is to evaluate the similarity between the subgraphs that each network contains. Network motif fingerprints [13] and graphlet-based metrics [14] are

- *D. Aparício, P. Ribeiro and F. Silva are with CRACS & INESC-TEC and the Department of Computer Science, Faculty of Sciences of the University of Porto, Porto 4169-007, Portugal*
*E-mail: {daparicio, pribeiro, fds}@dcc.fc.up.pt*

possible choices for this task. Both approaches compute the frequency of a set of small non-isomorphic subgraphs (also called *subgraph census*) but, in addition to that, graphlets also evaluate the contribution of each individual node from the network, producing a graphlet degree distribution that can be seen as an extension of the node degree concept. Furthermore, enumerating graphlets is computationally less expensive than calculating network motifs since the subgraphs are only enumerated in the original network, as opposed to also having to compute their occurrences in a large set of random networks in order to assess motif significance [15]. Graphlet-based metrics have been used to analyze networks from various biological areas such as protein-protein interaction [16], disease genes [17], age-related genes [18] or brain networks [19] due to the great amount of topological information that they provide [20].

Graphlet usage is often restricted to analyzing only the set of 30 undirected graphs of up to five nodes originally presented by [14] due to computational limitations. However, a substantial gain in topological information might be attained by examining different sets of graphlets. One possibility is to enumerate larger graphlets since, by definition, they capture more topological information about the network's structure than smaller graphlets, and this added information might be valuable. For instance, Hulovatyy et al. [21] observed that larger graphlets of 6 or 7 nodes led to an higher accuracy for node classification in dynamic networks than smaller ones. Additionally, in directed networks, edge direction should be taken into account since it can potentially reveal information about the network's structure that undirected graphlets are not able to capture. In this work we propose a novel extension of the graphlet methodology to directed networks and show how this may retrieve relevant information from biological networks. Despite the fact that graphlets are a general model, to the best of the authors' knowledge the only extensions to the original concept are relative to ordered [22] and dynamic graphlets [21]. The latter work by Hulovatty also goes beyond the usual 5-node graphlets (up to size 7).

There are many cellular biological networks that are intrinsically directed such as metabolic, cell signaling and gene transcriptional regulation networks. Methods and metrics that ignore the edge direction of these networks might be losing important information. Garlaschelli and Lofredo [23] proposed a new measure ($\rho$) to calculate the *link reciprocity* of a network that can be used to assess if its edge direction is important or not. Their measure is an absolute quantity ranging from $-1$ (*no reciprocity*) to 1 (*completely reciprocal*). Networks with a $\rho$-value of $\approx -1$ are *purely directional networks* meaning that edge direction is an intrinsic aspect of these networks and removing it makes them meaningless. On the other hand, networks with $\rho \approx 1$ can be safely transformed into topologically equivalent undirected networks without losing much information since their edges are always reciprocally connected. Garlaschelli and Loffredo calculated that celular and food web networks rank closer to the middle of the scale ($\rho \approx 0$) meaning that edge direction in these networks is significant. Additionally, some specific small directed graphs, such as feed-forward loops, have been shown to play a fundamental role in the organization of distinct networks [24].

Network motifs have been extensively used to study directed biological networks such as neural, transcriptional and signal networks [25]. Graphlets on the other hand are mostly restricted to undirected networks since, as mentioned previously, they consist of a set of undirected graphs. Park et al. [26] examined numerous directed biological networks using both directed motifs and undirected graphlets. Such a study could possibly benefit if a tool for enumerating directed graphlets was available.

In this article we present an efficient general-purpose tool to enumerate and compare both directed and undirected graphlet degree distributions. Furthermore, our tool can be used to enumerate arbitrarily large subgraphs (as long as they fit into memory) since it is not targeted for specific graphlets. Previous approaches either restricted the application of graphlets to undirected networks or had to ignore edge direction in directed networks, in practice reducing them to undirected networks. To achieve these objective we extend both i) the original concept of graphlets to *directed graphlets* and ii) upgrade a tree data-structure specialized in efficiently storing graphs, the g-trie, to a *graphlet-trie*. Our tool, GT-Scanner, can thus be used to enumerate directed and undirected graphlets as well as network motifs.

## 2 Materials and Methods

### 2.1 Graph and graphlet terminology

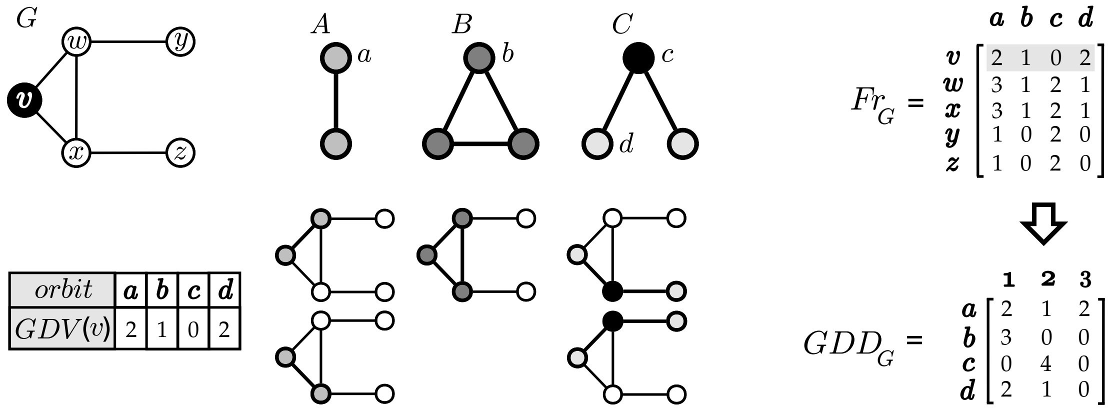
A network or *graph* $G$ is comprised of a set $V(G)$ of *vertices* or *nodes* and a set $E(G)$ of *edges* or *connections*. Nodes represent entities and edges correspond to the relationships between them. Edges are represented as pairs of vertices of the form $(a, b)$, where $a, b \in V(G)$. In *directed* graphs, edges $(i, j)$ are *ordered pairs* (translated to "$i$ *goes to* $j$") whereas in *undirected* graphs there is no order since the nodes are always reciprocally connected.

A $subgraph$ $G_k$ of $G$ is a graph of size $k$ where $V(G_k) \subseteq V(G)$ and $E(G_k) \subseteq E(G)$. A subgraph is $induced$ if $\forall u, v \in V(G_k) : (u, v) \in E(G_k)$ iff $(u, v) \in E(G)$. A $match$ or $occurrence$ of $G_k$ happens when $G$ has a set of nodes that induce $G_k$. Two matches are considered distinct if they have at least one different vertex. The $frequency$ of $G_k$ in $G$ is the number of occurrences of $G_k$ in $G$.

Two graphs are said to be *isomorphic* if it is possible to obtain one from the other just by changing the node labels without affecting their topology. All occurrences of a set $\mathcal{G}$ of non-isomorphic subgraphs must be enumerated in the original network before graphlet or network motif metrics can be calculated. We call this task the general subgraph census problem [27] and state it in Definition 2.1.

***Definition 2.1 (Subgraph Census Problem).*** Given a set $\mathcal{G}$ of non-isomorphic subgraphs and a graph $G$, determine the frequency of all induced occurrences of the subgraphs $G_s \in \mathcal{G}$ in $G$. Two occurrences are considered different if they have at least one node or edge that they do not share. Other nodes and edges can overlap.

*Graphlets* [14] are small induced non-isomorphic subgraphs structurally equivalent to *network motifs* [7] but that also include information about the position or *orbit* that nodes occupy in the graphlet. For instance, a node in the center of a star is different from the nodes on its periphery



| *orbit* | **a** | **b** | **c** | **d** |
|---|---|---|---|---|
| $GDV(v)$ | 2 | 1 | 0 | 2 |

$$Fr_G = \begin{array}{c|cccc} & \mathbf{a} & \mathbf{b} & \mathbf{c} & \mathbf{d} \\ \hline \mathbf{v} & 2 & 1 & 0 & 2 \\ \mathbf{w} & 3 & 1 & 2 & 1 \\ \mathbf{x} & 3 & 1 & 2 & 1 \\ \mathbf{y} & 1 & 0 & 2 & 0 \\ \mathbf{z} & 1 & 0 & 2 & 0 \end{array}$$

$$GDD_G = \begin{array}{c|ccc} & \mathbf{1} & \mathbf{2} & \mathbf{3} \\ \hline \mathbf{a} & 2 & 1 & 2 \\ \mathbf{b} & 3 & 0 & 0 \\ \mathbf{c} & 0 & 4 & 0 \\ \mathbf{d} & 2 & 1 & 0 \end{array}$$

**Fig. 1:** $GDV(v)$ obtained by enumerating the induced occurrences of all undirected graphlet orbits of sizes 2 and 3 ($A$, $B$ and $C$) touching $v$, and resulting $Fr_G$ and $GDD_G$ matrices for the complete subgraph census ($GDV(v)$ is highlighted in gray in $Fr_G$).

(see $G_{11}$ from Figure 4). The set of all orbits of $\mathcal{G}$ is referred to as $\mathcal{O}$. Another difference between graphlets and networks motifs is that the latter require a null model to verify if the subgraphs are overrepresented. Usually the null model is an artificial random network that maintains the original network's node degree sequence. On the other hand, graphlets do not require a null model and use the information of all subgraphs to perform a full-scale network comparison. The following graphlet definitions were first introduced in [14]. To compute the graphlet degree distribution it is necessary to count $\forall u \in V(G)$ how many times $u$ appears in some orbit $j \in \mathcal{O}$ and repeat this process for the total $m$ orbits, resulting in a graphlet degree vector $GDV(u)$ with $m$ positions. A matrix $Fr_G$ of $n \times m$ is obtained by joining the $GDV$s of all $n$ nodes where each row of $Fr_G$ is $GDV(v), v \in V(G)$ and each position $fr_{u,j}$ is the number of times that node $u$ appears in orbit $j$. This task is more formally defined in Definition 2.2.

$$Fr_G = \begin{bmatrix} fr_{0,0} & fr_{0,1} & \cdots & fr_{0,m} \\ fr_{1,0} & fr_{1,1} & \cdots & fr_{1,m} \\ \vdots & \vdots & \ddots & \vdots \\ fr_{n,0} & fr_{n,1} & \cdots & fr_{n,m} \end{bmatrix}$$

***Definition 2.2 (Orbit Frequency Computation).*** Given a set $\mathcal{G}_s$ of non-isomorphic subgraphs of size $s \in \{2, ..., k\}$ and a graph $G$, determine the number of times $fr_{i,j}$ that each node $i \in V(G)$ appears in all the orbits $j \in \mathcal{O}_s$. All occurrences are induced. Two occurrences are considered different if they have at least one node or edge that they do not share. Other nodes and edges can overlap.

Both directed and undirected graphlets are considered in this work; $d\mathcal{G}_k$ and $d\mathcal{O}_k$ denote the set of all directed graphlets and directed orbits of size $s \in \{2, ..., k\}$, respectively, while $u\mathcal{G}_k$ and $u\mathcal{O}_k$ are used for the undirected counterparts. Therefore, a full graphlet enumeration of size $k$ not only counts all graphlets of size $k$ but also the graphlets of size $< k$. Partial occurrences are not counted since graphlets are induced subgraphs. For instance, triangle $B$ partially matches three chains $C$ which are not considered since they do not constitute a full match. Sometimes network motifs are used as partial occurrences [28] however, like graphlets, researchers use them more commonly to account only for induced occurrences [7], [29]. Contrarily to partial occurrences, induced occurrences are not ambiguous because existing and non-existing edges are given equal importance, leading to more revealing and less convoluted results.

Matrix $Fr_G$ is transformed into a graphlet degree distribution $GDD_G$ where $d_G^j(k)$ denotes how many nodes appear $k$ times in orbit $j$. $GDD_G$ offers detailed topological information of $G$ and it is the output of our approach when performing a census on a single network. An illustration of how the $GDV$ of a vertex $v$ is computed for $u\mathcal{O}_3$ is presented in Figure 1, as well as the resulting $Fr_G$ and $GDD_G$ matrices for the small network.

$$GDD_G = \begin{bmatrix} d_G^0(1) & d_G^0(2) & \cdots & d_G^0(+\infty) \\ d_G^1(1) & d_G^1(2) & \cdots & d_G^1(+\infty) \\ \vdots & \vdots & \ddots & \vdots \\ d_G^m(1) & d_G^m(2) & \cdots & d_G^m(+\infty) \end{bmatrix}$$

Two networks $G$ and $H$ can then be compared by computing the differences between their respective $GDD$ matrices after their distributions are normalized - represented below as $n_G^j(k)$. In our experiments the arithmetic mean GDD-agreement ($GDA$) introduced by [14] is used, defined as follows.

$$GDA(G,H)^j = 1 - \frac{1}{\sqrt{2}} \left( \sum_{k=1}^{+\infty} [n_G^j(k) - n_H^j(k)]^2 \right)^{\frac{1}{2}} \quad (1)$$

$$GDA(G,H) = \frac{1}{m} \sum_{j=0}^{m} GDA(G,H)^j \quad (2)$$

When two (or more) networks are compared our method outputs $GDA(G,H)$ as well as $GDD_G$ and $GDD_H$. A high $GDA(G,H)$ means that $G$ and $H$ are topologically similar. Since both directed and undirected $GDA$s can be calculated, $uGDA_k$ and $dGDA_k$ represent the $GDA$s when comparing undirected and directed graphlets, respectively, of size $k$. We adjusted the metric to only consider orbits that appear in at least one of the networks; the original metric is henceforth referred to as $GDA'$ and our own as $GDA$. Modifying the metric was necessary since non-appearing orbits would contribute to unreasonably high $GDA'$s when enumerating a large number of orbits or when

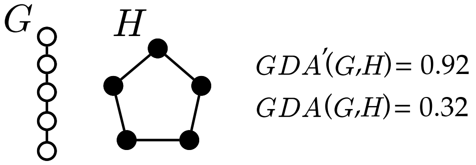


**Fig. 2:** Comparison of different $GDA$ metrics. The original metric $GDA'$ was found to produce unreasonably high values for small networks (in the example) or when many graphlets are enumerated. This makes the metric inappropriate to compare directed graphlets since the number of orbits is very high. In our modified $GDA$ metric only orbits appearing in either $G$, $H$ or both are considered, discarding non-present orbits.

small networks were used. This happens because the $GDA'$ of two networks is increased even if the orbit frequency is zero in both networks. Hulovatty et al. [21] also considered graphlets with more than the usual 5 nodes and suggested a similar explanation. This is not very problematic when a small number of orbits is enumerated, such as the original 73 undirected ones; however, as can be seen from Table 1, bigger undirected graphlets and directed graphlets may require thousands or millions of orbits to be enumerated. For these cases it is likely that many of the possible orbits do not appear in either network, which may result in higher $GDA'$s than expected. Tests performed on small food webs produced an average $GDA'$ of $\approx 0.5$ when enumerating $d\mathcal{G}_4$, and it increased to $\approx 0.85$ for $d\mathcal{G}_5$. This does not translate to those food webs being much more alike when looking at their larger graphlets but rather that there were many orbits that did not appear in either network. Figure 2 illustrates the difference in agreement values given by the original $GDA'$ metric and our own. Another problem lies in the huge number of possible orbits ($|\mathcal{O}_k|$) that directed graphlets with more than 5 nodes have ($k > 5$). For instance, when $k = 6$ there are more than 1 million potential orbits. Assuming that the frequency of each one is stored in an 4-byte integer, computing $\mathcal{O}_k$ would require $|V(G)| \times 4 \cdot |\mathcal{O}_k|$ bytes of memory. Therefore, enumerating $d\mathcal{G}_5$ on a network with $10^5$ nodes would take $\approx 4GB$ of RAM, which is still feasible in most modern PCs; however, $d\mathcal{G}_6$ would only be possible for networks with a few hundred nodes and if $k \geq 6$ the enumeration would simply not be viable. A possible way to reduce the memory footprint is to deal with orbit redundancies [30]. Nonetheless, larger values of $k$ still produce too many non-redundant orbits that make the computation infeasible in terms of memory. Another option is to avoid generating all possible graphlets and orbits before the enumeration and instead build their representation during the enumeration phase as they occur in the network [31] since it is reasonable to expect that only a fraction of all possible graphlets/orbits actually appear in a given network. This strategy may introduce an overhead in computational time but makes it attainable to analyse larger graphlets.

## 2.2 The G-Trie Data-structure

A g-trie [32] is a tree-like data-structure created initially to calculate network motifs but it can be efficiently used to solve the general subgraph census problem. Ribeiro and Silva [29] presented a g-trie algorithm that was one or two orders of magnitude faster than previous approaches, and g-tries are still state-of-the-art for network motif discovery efficiency-wise.

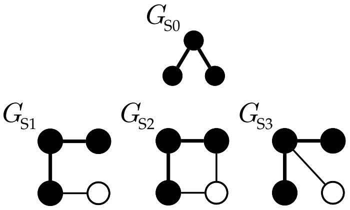


**Fig. 3:** Common topology of three graphs. G-Tries use common structures between the graphs of $\mathcal{G}_s$ to heavily constrain the search space.

The efficiency of the g-trie data-structure is mostly due to two main algorithmic ideas. First, the search space is heavily constrained by identifying common subtopologies between the set of subgraphs $\mathcal{G}_s$ before enumerating them. Figure 3 illustrates this base concept by showing three small graphs that share a common subtopology. In practice this means that instead of enumerating each subgraph, $G_{S1}$, $G_{S2}$ and $G_{S3}$, individually, a g-trie starts by looking for occurrences of the smaller common subgraph $G_{S0}$ and then performs the necessary expansions for each larger subgraph. Secondly, symmetry breaking conditions are automatically generated to eliminate automorphisms, thus avoiding redundancies and guaranteeing that each occurrence is found only once. For instance, without these conditions an $n$-clique ($n$ nodes fully connected to each other) would be found $n!$ times (once for each possible permutation of the $n$ nodes) since all these permutations would be a match to a g-trie path from the root to the node representing that $n$-clique.

A g-trie receives as input the list of graphs that the user wants to enumerate, which can be all undirected graphlets with up to five nodes, a set of directed graphs, specific interesting patterns (such as cliques or stars), or any other desired graphs. However, due to the nature of the g-trie, which relies on common subtopologies between graphs to construct a compact search tree, g-tries are better suited for tasks where one wants to count the occurrences of *many small graphs* inside a large network. G-Tries can be fully constructed before the enumeration and stored in a file or built on-the-fly to avoid having to store all possible graphs.

*Graphlet-tries* are an extension of g-tries that also consider the nodes' orbit. The broader term *g-trie* is used whenever a concept applies to both g-tries and graphlet-tries. A graphlet-trie containing all the original 30 graphlets is shown in Figure 4. Notice that this representation is not unique since there exist many possible ways to represent the same graph by modifying the vertex labels. Therefore, before creating a g-trie it is necessary to establish a canonical form that efficiently compresses the search space by finding as many common subtopologies in $\mathcal{G}_s$ as possible. G-Tries also need mechanisms to avoid counting isomorphic graphs more than once that are not discussed here. The original orbit numbers from [14] are kept only for convenience since they are generated automatically in our implementation. All 2 and 3-node directed graphlets are illustrated in Figure 5 as well as the non-bidirectional 4-node directed graphlets (the bidirectional graphlets were removed for space concerns). An additional graphlet-trie containing them is is presented in the Supplementary Material.

| | $u\mathcal{G}_k$ | | $d\mathcal{G}_k$ | |
|---|---|---|---|---|
| $k$ | $\lvert\mathcal{G}_k\rvert$ | $\lvert\mathcal{O}_k\rvert$ | $\lvert\mathcal{G}_k\rvert$ | $\lvert\mathcal{O}_k\rvert$ |
| 2 | 1 | 1 | 2 | 3 (1.5 $\times$ $\lvert\mathcal{G}_k\rvert$) |
| 3 | 3 | 4 | 15 | 33 (2.3 $\times$ $\lvert\mathcal{G}_k\rvert$) |
| 4 | 9 | 15 | 214 | 730 (3.5 $\times$ $\lvert\mathcal{G}_k\rvert$) |
| 5 | 30 | 73 | 9,578 | 45,637 (4.8 $\times$ $\lvert\mathcal{G}_k\rvert$) |
| 6 | 142 | 480 | 1,540,421 | 9,121,657 (5.9 $\times$ $\lvert\mathcal{G}_k\rvert$) |
| 7 | 965 | 4,786 | 882,011,563 | $\approx 7 \times \lvert\mathcal{G}_k\rvert$ |
| 8 | 12,082 | 77,275 | 1,793,355,966,869 | $\approx 8 \times \lvert\mathcal{G}_k\rvert$ |
| 9 | 273,162 | 2,188,288 | 13,027,955,038,433,121 | $\approx 9 \times \lvert\mathcal{G}_k\rvert$ |

**TABLE 1:** Number of undirected and directed graphlets, as well as their respective orbits, depending on the size of the graphlets. For each case, all graphlets of sizes $2..k$ are counted. It is impractical to enumerate all possible orbits for $d\mathcal{G}_k$ when $k$ is larger than 6 due to the size of $|\mathcal{O}_k|$.

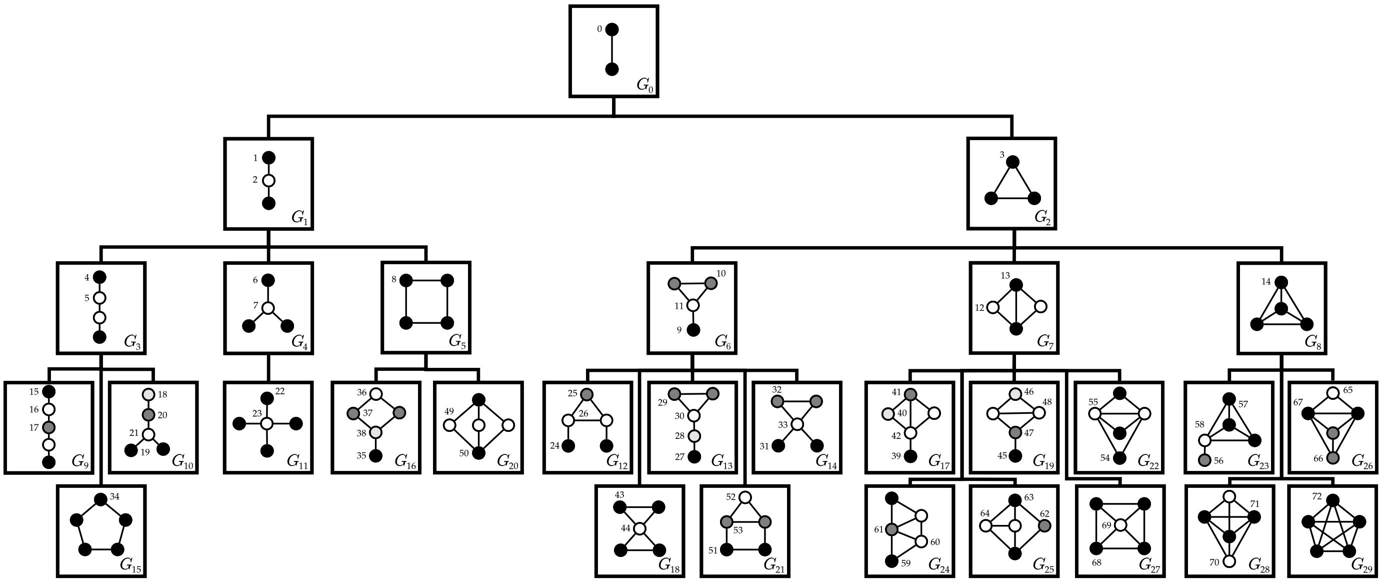


**Fig. 4:** A graphlet-trie containing all 2, 3, 4 and 5-node undirected graphlets $G_0, ..., G_{29}$ as they were presented in [14]. In a g-trie the common topologies between graph(let)s of different sizes become evident.

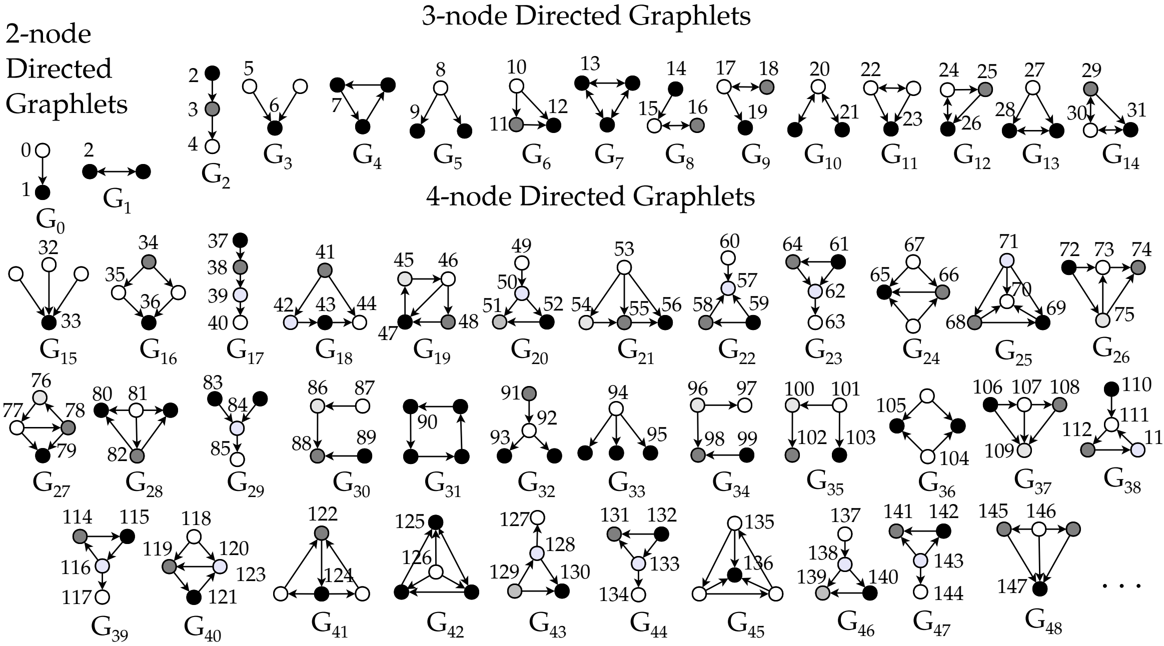


**Fig. 5:** A subset of $d\mathcal{G}_4$ containing all 2 and 3-node directed graphlets and the 4-node directed graphlets that have no bidirectional edges (for space concerns).

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCBB.2016.2586046, IEEE/ACM Transactions on Computational Biology and Bioinformatics

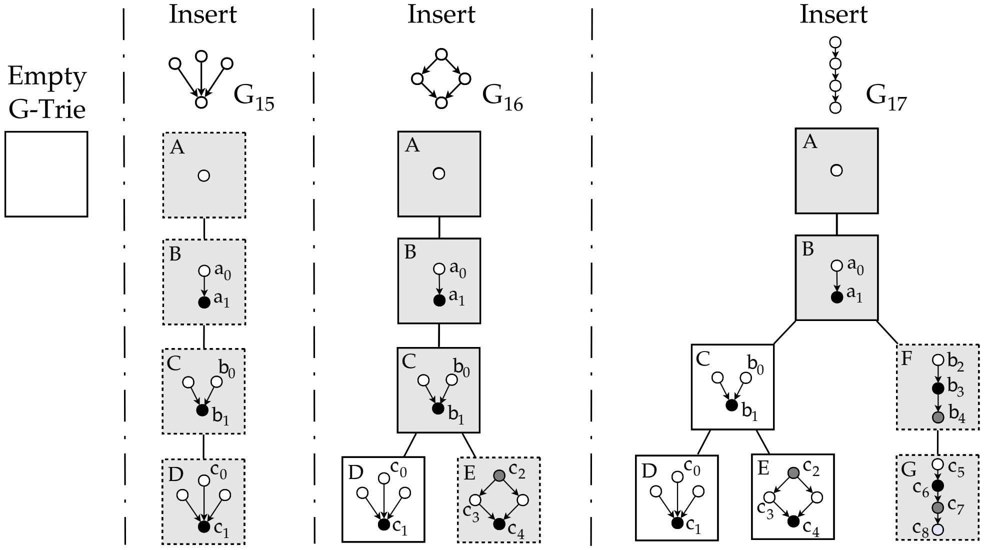


**Fig. 6:** Iterative insertion of 3 graphlets to a graphlet-trie. Graphlet-trie vertices colored in gray represent the path to the newly added graphlet. Vertices with dotted contours were added in the most recent insertion.

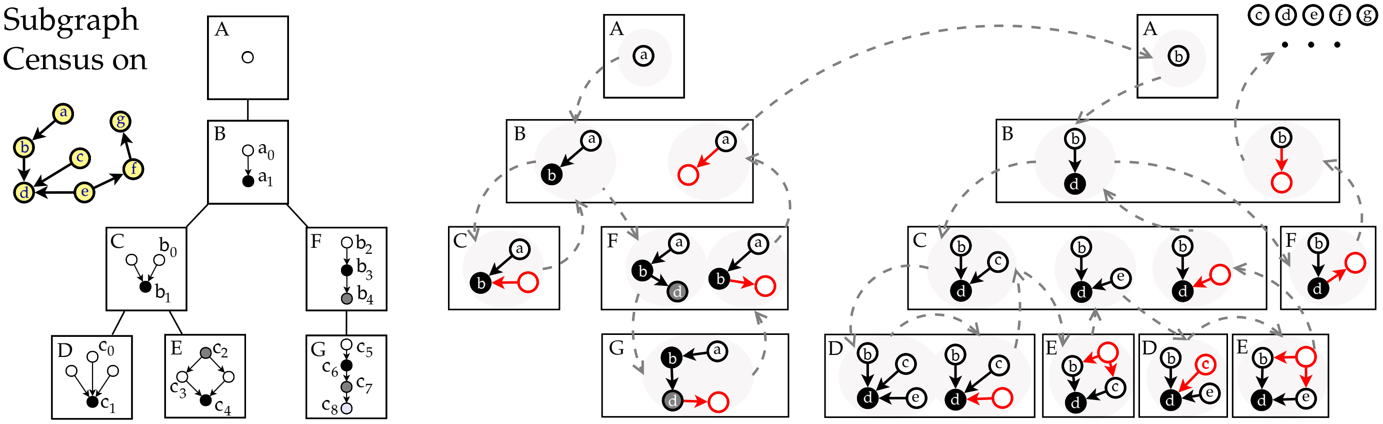


**Fig. 7:** Subgraph census using a graphlet-trie. Nodes in red mean that no candidate was found for that particular graphlet. The dotted arrows represent the search path.

### 2.2.1 G-Trie Creation

**Algorithm 1** Populate a g-trie $T$ with subgraphs $G \in \mathcal{G}$

```
1:  procedure CreateGTrie($\mathcal{G}$)
2:      $T \leftarrow$ EmptyGTrie()
3:      for all $G \in \mathcal{G}$ do
4:          Insert($T.root, G, 1$)
5:  procedure Insert($N, G, depth$)
6:      for all $C \in N.children$ do
7:          if ShareCommonTopology($C, G, depth$) then
8:              Insert($C, G, depth + 1$)
9:              return
10:     $NewChild \leftarrow N$.addChild($G, depth$)
11:     Insert($NewChild, G, depth + 1$)
```

Figure 6 shows how a g-trie is built by iterative insertion and Algorithm 1 presents the respective pseudo-code for the task. When graph $G_{15}$ is inserted into an initially empty g-trie (lines 2-4) no common subtopologies are found (lines 6-7) and four new g-trie nodes need to be created (lines 10-11), $A$, $B$, $C$ and $D$, each containing a subgraph of size equal to the depth level of the g-trie ($A$ has 1 node, $B$ has 2 nodes, and so forth). A graphlet-trie also evaluates the subgraph orbits and stores them alongside the graphlet. When $G_{16}$ is inserted to the tree, only node $E$ needs to be added to it since $G_{15}$, which was previously inserted, and $G_{16}$ share the common path $A \Rightarrow B \Rightarrow C$ in the graphlet-trie (lines 6-9). Finally, $G_{17}$ requires two nodes to be created, $F$ and $G$, because it only shares the path $A \Rightarrow B$ with the other two graphlets.

In this very short example, the compression rate achieved by the g-trie is $1 - \frac{7}{12} \approx 42\%$ since each 4-node graphlet would require 4 different subgraphs to be queried (one for each $k \in \{1, 2, 3, 4\}$, giving a total of 12 for the 3 subgraphs) but, by using the common topology of the subgraphs, only 7 are actually needed by the g-trie. Inserting all 4-node graphlets gives a higher compression ratio of $\approx 80\%$ since there are more opportunities for the g-trie to find common topologies between the subgraphs [33]. Usually graphlet enumeration requires a census of not only size $k$ but also all $s < k$. This can be achieved by first creating a g-trie of size $k$ and then marking all other non-isomorphic subgraphs of sizes smaller than $k$ that appear on the g-trie. The set of subgraphs that the user wants to query on the network is fully customizable and given as input, making g-tries a very flexible approach.

### 2.2.2 Subgraph Census with a G-Trie

**Algorithm 2** Count all orbits from g-trie $T$ in network $G$

```
 1: procedure COUNTALL($T, G$)
 2:     for all vertex $v$ of $G$ do
 3:         for all children $c$ of $T.root$ do
 4:             COUNT($c$, $\{v\}$)
 5: procedure COUNT($T, V_{used}$)
 6:     $T$.ORBITS($V_{used}$)++
 7:     $V \leftarrow$ GETVALIDCANDIDATES($T, V_{used}$)
 8:     for all vertex $v$ of $V$ do
 9:         for all children $c$ of $T$ do
10:             COUNT($c, V_{used} \cup \{v\}$)
11: function GETVALIDCANDIDATES($T, V_{used}$)
12:     $V_{conn} \leftarrow$ vertices connected to the vertex being added
13:     $m$      $\leftarrow$ vertex of $V_{conn}$ with smallest neighborhood
14:     $V_{cand} \leftarrow$ neighbors of $m$ that respect both
15:              connections to ancestors and
16:              symmetry breaking conditions
17:     return $V_{cand}$
```

Figure 7 illustrates how the previously created g-trie from Figure 6 is used to perform subgraph census. The g-trie search algorithm is essentially a depth-first search mapping the input network to the g-trie. The notation $(N, \{v_1, v_2, ..., v_k\})$ represents vertices $v_1, v_2, ..., v_k$ from the input network which are mapped to node $N$ on the g-trie. For simplicity, $N$ can refer to either the g-trie node or its respective subgraph.

The network from Figure 7 has seven vertices which are all mapped one at a time to $A$, the initial node of the g-trie (lines 2-4 of Algorithm 2). Starting from $(A, \{a\})$ the search descends on the g-trie and looks for valid candidates (line 7). For an efficient graph traversal, our algorithm picks the vertex from $V_{used}$ that is connected to the newly added node in g-trie $T$ (line 12) that has the smallest neighborhood (line 13). At the beginning $A$ is the only possible choice. The candidate vertices $c \in V_{cand}$ are the neighbors of $A$ that respect both the g-trie connections (in this case, candidate $c$ only needs to have an incoming edge from $A$) and symmetry breaking conditions, which are needed to avoid isomorphic cases (lines 14-17). The first viable candidate is vertex $B$, and the algorithm finds the mapping $(B, \{a, b\})$, incrementing the frequency of orbits $a$ and $b$ (line 6). The search continues in depth-first fashion (lines 8-10) but no valid mapping exists between $C$ and $\{a, b, v_i\}$ since no $v_i$ can be joined to $\{a, b\}$ so that the resulting subgraph forms $C$. Therefore, the algorithm backtracks to $(B, \{a, b\})$ and instead looks for a valid mapping of $F$, finding $(F, \{a, b, d\})$. No valid mapping is found for $(G, \{a, b, d, v_i\})$ and when the search backtracks no further alternatives for $(F, \{a, b\})$ are discovered. Since no other valid mapping are found for $(B, a, v_i)$, the algorithm moves on to vertex $b$. The census finds $(A, \{b\})$, $(B, \{b, d\})$, $(C, \{b, d, c\})$ and $(D, \{b, d, c, e\})$ until it has to backtrack since there are no more alternatives for $(D, \{b, d, c, v_i\})$. There are also no occurrences of $(E, \{b, d, c, v_i\})$ so the algorithm proceeds and finds $(C, \{b, d, e\})$. When the algorithm reaches $D$, $(D, \{b, d, e, c\})$ is a valid mapping topologically, however it would be the same occurrences as the previously found $(D, \{b, d, c, e\})$. In practice, g-tries do not find repeated occurrences thanks to *symmetry breaking* mechanisms embedded in the g-trie nodes (line 16): vertices that appear later in the same orbit are only valid if they have a bigger index than the previous vertices of the same orbit. After failing to find a valid mapping for $(D, \{b, d, e, v_i\})$ the search also fails for $(E, \{b, d, e, v_i\})$ and keeps backtracking until it proceeds to $(A, \{c\})$. For more specific details on how a g-trie is created and used for subgraph enumeration the reader is referred to [32]. Additionally, g-tries have already been extended to also support weighted [34] and colored [35] networks. A tool for motif discovery making use of the g-trie datastructure has already been released[1]. Our tool extends the previous one by also allowing for graphlet enumeration and network comparison.

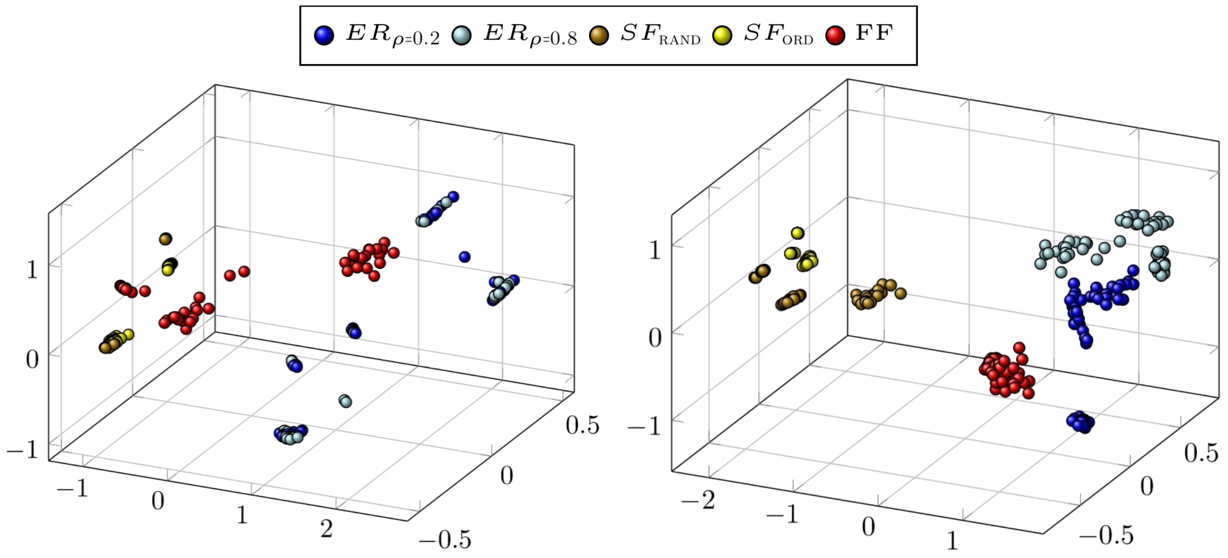
## 3 Results and Discussion

To assess the applicability and performance of our tool, we organise the set of experiments into two parts: i) classification accuracy on synthetic data and ii) performance evaluation on real biological data. The first measures how well directed graphlets can group a set of directed networks and compares it with undirected graphlets, while the latter analyses the performance of our tool on a set of directed biological networks of different types (biological function) by comparing its execution time with state-of-the-art approaches.

### 3.1 Classification Accuracy on Synthetic Networks

In order to evaluate the advantages of using directed graphlets we measure how well they group networks of different types and compare their results with those produced by undirected graphlets. These tests were performed on synthetic networks pertaining to different graph types: Erdős-Rényi random graphs (ER) [36], scale-free networks (SF) [37] and Forest Fire graphs (FF) [38]. The ER networks are generated as regular ER graphs but, since they are directed, it is also necessary to choose the direction of each edge. Two groups of ER networks, $ER_{\rho=0.2}$ and $ER_{\rho=0.8}$, were created with different probabilities $\rho$ for nodes to be reciprocally connected: $\rho = 20\%$ and $\rho = 80\%$. The non-reciprocal edges of $v$ (for either group) have an equal chance of being an *in edge* $(u, v)$ or an *out edge* $(v, u)$. While the networks of these two groups are very similar if one disregards their edge direction they vary greatly when the direction of the edges is considered, and an appropriate metric should be able to distinguish them. We proceed by generating undirected scale-free networks with the same exponent $\gamma$ and divide them into two directed groups: $SF_{\text{ORD}}$, where the undirected edges $(u, v)$ are transformed into a similar directed edge $u \to v$ where $u < v$, and $SF_{\text{RAND}}$ where the direction is randomly assigned. In this way we obtain networks that have a very similar node degree distribution but very different *in* and *out* node degree distributions. All

[1] http://www.dcc.fc.up.pt/gtries/



(a) $u\mathcal{G}_4$

(b) $d\mathcal{G}_4$

**Fig. 8:** MDS representation applied to the $GDA$ matrices obtained for undirected graphlets $u\mathcal{G}_4$ and directed graphlets $d\mathcal{G}_4$. (a) Undirected graphlets can not appropriately cluster the different networks whereas (b) directed graphlets clearly group the networks correctly.

ER and SF graphs have $\approx 1\%$ edge density, mimicking real world networks. Finally, the FF networks have $p = 0.37$ and $p_b = 0.32$, as suggested by [38] in order to build the most realistic networks. For each of the aforementioned types we generated 20 networks with 500, 1000 or 2000 nodes, giving a total number of $5 \times 20 \times 3 = 300$ networks. The $GDA$ was computed for all pairs of networks for three distinct sets of graphlets: $u\mathcal{G}_4$, $d\mathcal{G}_4$ and $d\mathcal{G}_5$. The clustering capabilities of both directed and undirected graphlets are illustrated using multidimensional scaling (MDS) [39] in a 3-dimensional space. The performance of each set of graphlets is evaluated by comparing their obtained precision-recall when classifying the networks. This methodology was previously adopted by [40] to demonstrate how well their metric (GCD) could group different undirected networks using undirected graphlets.

Figure 8 shows the MDS embeddings of the 300 networks using (a) $u\mathcal{G}_4$ and (b) $d\mathcal{G}_4$. Undirected graphlets successfully distinguish between different directed graph models ($ER$ vs $SF$ vs $FF$) despite ignoring their edge direction. This is possible since the topology of the different models is so distinct that edge direction can be disregarded. However, as expected, when the undirected topology of the networks is similar it is necessary to take the edge direction into account. Directed graphlets successfully separated networks of different models ($ER$ vs $SF$ vs $FF$) and also accurately grouped networks of different types; they separated networks with different levels of reciprocity ($ER_{\rho=0.2}$ vs $ER_{\rho=0.8}$) and with distinct in/out-degree distributions ($SF_{\text{ORD}}$ vs $SF_{\text{RAND}}$). Furthermore, undirected graphlets inadequately separated the FF networks by size while directed graphlets clustered them together. The precision-recall curves for undirected ($u\mathcal{G}_4$) and directed graphlets ($d\mathcal{G}_4$ and $d\mathcal{G}_5$) shown in Figure 9 were obtained by computing the $GDA$ distance between each pair of networks and calculating if that distance is smaller than a threshold $\epsilon$ and, if it was, the networks were grouped together. Since the $GDA$ varies from 0 to 1, initially $\epsilon$ is set as 0 (meaning that the networks are exactly the same according to the metric) and is incremented by $s = 0.001$ at each step. Precision is the fraction of correctly grouped pairs while recall is the fraction of the correctly grouped pairs over all correct ones. The Area Under the Precision-Recall curve (AUPR) evaluates how well the metric groups the networks, and its value is approximated as shown in Equation 3. $Pr(k)$ is the precision at step $k$, $\Delta Rec(k)$ is the change in recall from steps $k-1$ to $k$ and $N$ is the number of steps.

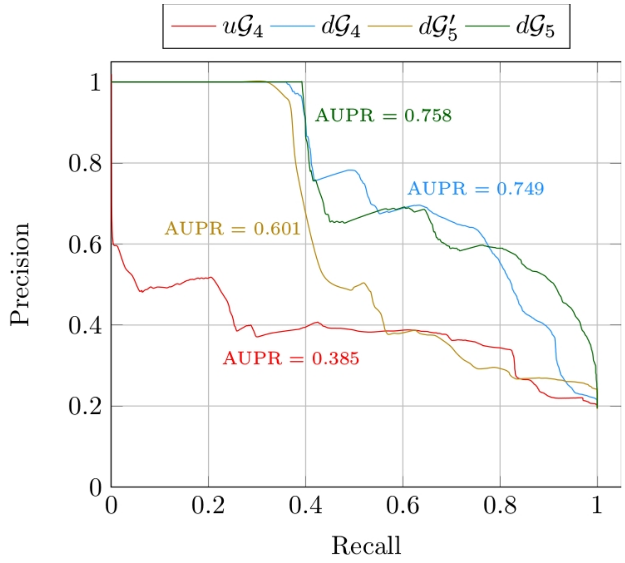


**Fig. 9:** Precision-recall curves for undirected graphlets ($u\mathcal{G}_4$) and directed graphlets ($d\mathcal{G}_4$, $d\mathcal{G}_5$ and $d\mathcal{G}'_5$). Undirected graphlets can not correctly group the networks (AUPR = 0.385). Directed graphlets ($d\mathcal{G}_4$ and $d\mathcal{G}_5$) correctly cluster the networks (AUPR $\approx$ 0.75). Non-appearing orbits undermine larger graphlets' ($d\mathcal{G}'_5$) capability to cluster the networks (AUPR = 0.601).

$$AUPR = \sum_{k=1}^{N=1000} Pr(k)\Delta Rec(k) \tag{3}$$

The AUPR for $u\mathcal{G}_4$ is 0.385, which is clearly lower than the one obtained by $d\mathcal{G}_4$ (0.749). Two sample Welch t-tests were performed to verify that the results for directed graphlets were statistically significant (p-values $\leq$ 0.01). The effect of removing non-appearing orbits from the $GDA$ computation was also measured; when non-appearing orbits were not removed from the $GDA$ computation ($d\mathcal{G}_5'$) the AUPR was relatively low (0.601). This negative effect is more severe when larger graphlets are enumerated because, as discussed in Section 2, the resulting $GDA$s are always very high since there are many non-appearing orbits, contributing to a false similarity. Removing non-appearing orbits from the $GDA$ computation greatly increases the clustering capabilities of larger directed graphlets; $d\mathcal{G}_5$ obtained the highest AUPR value at 0.758 among all graphlet sets. These experiments show that directed graphlets can be successfully used to group directed networks pertaining to the same type and likewise distinguish between directed networks of different types.

It could be argued that directed graphlets are not necessary to correctly cluster these types of networks since one could simply analyse their reciprocity and/or the degree distribution to be able to separate them. In these experiments our main concern was to show that undirected graphlets are not suitable to study directed networks since, by definition, they can not even differentiate networks that differ on simple directed metrics. However, directed graphlets offer much richer topological information than reciprocity or degree distributions, in the same way that undirected graphlets give much more details on the network's structure than just the node-degree. In fact, reciprocity and degree-distribution are embedded in the two 2-node directed graphlets in the same way that the undirected node degree is represented by the 2-node undirected graphlet. Enumerating directed graphlets of more than 2-nodes captures not only these two basic metrics but also the more intricate connections between the nodes.

Consider Figure 10: networks $G$ and $G'$ have the same level of reciprocity ($\frac{1}{3}$) but they represent different subgraphs since they are not isomorphic, while networks $G$ and $G''$ have the same in and out-degree distributions but they are completely different nevertheless. Therefore, reciprocity would not be sufficient to distinguish $G$ from $G'$ and the degree distribution could not separate $G$ from $G''$. Undirected graphlets could not distinguish $G$ from $G'$ but they could differentiate between $G$ and $G''$ since their undirected topology is not the same. On the other hand, directed graphlets are perfectly capable of distinguishing all three cases since they are more general than both a) undirected graphlets, b) reciprocity and c) degree distribution. In fact, directed graphlets combine all three metrics, providing a powerful way to analyse directed networks' topology and overcoming the limitations of undirected graphlets for such networks.

### 3.1.1 Computational Complexity

Taking the edge consideration into account augments the complexity of the already computationally demanding exhaustive subgraph enumeration. Furthermore, it might be interesting to enumerate graphlets (both directed and undirected) that have more than 5-nodes. However, searching for larger graphlets increases the execution time exponentially. Therefore, a very efficient tool is required for this task to be feasible. In the following section we compare our tool with other tools that perform exhaustive subgraph enumeration.

## 3.2 Performance on Real Biological Networks

### 3.2.1 Experimental Setting

Our experimental results were gathered on a 8-core machine consisting of two quad-core IntelÂő XeonÂő Processor E5620 processors at 2.4GHz with a total of 12GB of memory. Code for GT-Scanner was developed in C++11 and compiled using gcc 4.8.2. As noted previously, the version of GT-Scanner that only performed network motif discovery was released prior to this work. The other tools used for comparison were also developed in C++ and are available as open-source. The execution times of each tool are relative only to the graphlet enumeration phase, not taking into account the time taken to load the graph into memory nor perform other initialization and finalization tasks.

### 3.2.2 Directed Biological Networks

There are numerous kinds of intra-cellular networks, such as metabolic, transcriptional regulatory and cell signaling networks, where edge direction is intrinsically related to its function. Metabolic networks represent the set of biochemical reactions occurring within a cell that allow the organism to grow, reproduce, respond to the environment, and other biological functions essential for the organism's survival. These reactions are catalyzed by enzymes that act upon substrates. Therefore, in metabolic networks a node can be an enzyme or a substrate and the connections are directed edges going from enzymes to substrates. Transcriptional regulatory networks model the process by which the information in the genes is transcribed into proteins or RNA, also called gene expression. In these networks nodes are either transcription factors or proteins that are connected by directed edges representing how the transcription factors influence the gene by stimulating or repressing its expression. A cellular signaling network is comprised of a sequence of biochemical reactions between cells of the same organism. A great number of tasks such as the development, repair and immunity of cells depend on the proper functioning of cell signaling networks. Nodes in these networks are proteins and edges exist between activator and receptor proteins that communicate through signals from the first to the latter.

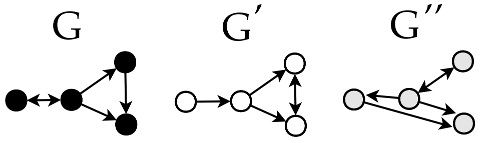
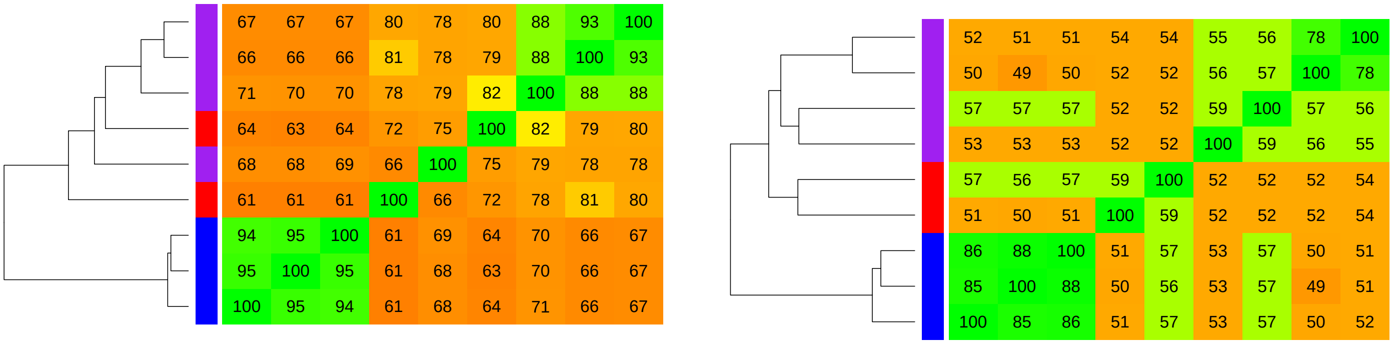


**Fig. 10:** Reciprocity or degree distribution information are not sufficient to distinguish two graphs. Directed graphlets incorporate both metrics and offer more detailed topological information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCBB.2016.2586046, IEEE/ACM Transactions on Computational Biology and Bioinformatics

| $G$ | SIG_NCI | SIG_NH | SIG_SH | SIG_SM | MET_BS | MET_DR | MET_TY | TR_EC | TR_YST |
|---|---|---|---|---|---|---|---|---|---|
| $\|V(G)\|$ | 15,533 | 1,634 | 529 | 477 | 453 | 2,280 | 2,361 | 99 | 688 |
| $\|E(G)\|$ | 23,682 | 4,665 | 1,223 | 1,056 | 2,025 | 5,588 | 5,822 | 212 | 1,078 |
| Type | Signaling | | | | Metabolic | | | Transcriptional | |
| Source | [41] | [42] | [43] | | [3] | | | [24] | [7] |

**TABLE 2:** Set of biological networks used for experimental evaluation: cell signaling, metabolic and transcriptional regulatory networks.



| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 67 | 67 | 67 | 80 | 78 | 80 | 88 | 93 | 100 |
| 66 | 66 | 66 | 81 | 78 | 79 | 88 | 100 | 93 |
| 71 | 70 | 70 | 78 | 79 | 82 | 100 | 88 | 88 |
| 64 | 63 | 64 | 72 | 75 | 100 | 82 | 79 | 80 |
| 68 | 68 | 69 | 66 | 100 | 75 | 79 | 78 | 78 |
| 61 | 61 | 61 | 100 | 66 | 72 | 78 | 81 | 80 |
| 94 | 95 | 100 | 61 | 69 | 64 | 70 | 66 | 67 |
| 95 | 100 | 95 | 61 | 68 | 63 | 70 | 66 | 67 |
| 100 | 95 | 94 | 61 | 68 | 64 | 71 | 66 | 67 |

(a) $uGDA_5$

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 52 | 51 | 51 | 54 | 54 | 55 | 56 | 78 | 100 |
| 50 | 49 | 50 | 52 | 52 | 56 | 57 | 100 | 78 |
| 57 | 57 | 57 | 52 | 52 | 59 | 100 | 57 | 56 |
| 53 | 53 | 53 | 52 | 52 | 100 | 59 | 56 | 55 |
| 57 | 56 | 57 | 59 | 100 | 52 | 52 | 52 | 54 |
| 51 | 50 | 51 | 100 | 59 | 52 | 52 | 52 | 54 |
| 86 | 88 | 100 | 51 | 57 | 53 | 57 | 50 | 51 |
| 85 | 100 | 88 | 50 | 56 | 53 | 57 | 49 | 51 |
| 100 | 85 | 86 | 51 | 57 | 53 | 57 | 50 | 52 |

(b) $dGDA_4$

**Fig. 11:** Heatmaps and dendrograms of the $uGDA_5$ (a) and $dGDA_4$ (b) obtained for the tested networks. Undirected graphlets accurately clustered the metabolic networks (blue) but incorrectly grouped cell signaling (purple) with transcriptional regulatory networks (red). Directed graphlets were able to cluster all networks by type without error.

### 3.2.3 Methodology

The computational networks used in these experiments, detailed in Table 2, are evidently a translation of real biological networks, thus potentially making the process of finding their similarities harder since their real structure may not be fully represented. Nevertheless, it is usually assumed that network structural similarity of computational networks may also indicate functional similarity [44]. Thus, one can expect that networks belonging to the same type to be more topologically similar than networks of different types.

In order to verify if that is the case for these specific networks we to assess their topological similarity in terms of their relative graphlet distributions. This comparison is performed for each pair of networks $(G, H)$ from Table 2 by first enumerating all graphlet orbits of both $G$ and $H$ and then comparing their $GDD$s by computing the $dGDA_4(G, H)$ and $uGDA_5(G, H)$. The results are shown for $d\mathcal{G}_4$ since 4-node directed graphlets were already successful in correctly clustering the networks by type, removing the need to look for larger subgraphs, and also for $u\mathcal{G}_5$ because they are the set of graphlets most often used in the literature.

The matrices obtained after computing $uGDA_5(G, H)$ and $dGDA_4(G, H)$, respectively, for each pair of networks are displayed in Figure 11 along with their corresponding dendrogram and heatmap. It can be observed that networks of the same type are correctly grouped by their $GDA$ using directed graphlets. This is an indicator that directed graphlets can detect topological similarities between real directed biological networks of the same type and can likewise find structural differences between networks of different types. Undirected graphlets did not correctly separate cell signaling from transcriptional regulatory networks. Again, since the networks come from distinct sources and noise is embedded in the data, it is not guaranteed that these networks are actually being separated by function and not by bias in their representation. Nevertheless, graphlets are a powerful tool to assess functional similarity and directed graphlets, by definition, capture more functional similarity in directed networks than undirected ones.

### 3.2.4 Performance Comparison

Currently, the most popular tool for graphlet discovery is GraphCrunch [45]. Observing its source code one notices that the possible 30 subgraphs are enumerated manually, therefore it is not possible to enumerate a different set of graphlets. Until recently it performed a full enumeration of all graphlets of up to size 5 in order to calculate their orbit frequency. A more recent tool, Orca [46], was shown to perform one or two orders of magnitude faster than the original GraphCrunch and has since been integrated into it. Henceforth, when GraphCrunch is referenced we are alluding to its version before adopting Orca's algorithm to perform the enumeration. Orca achieves its performance by observing that, given a limited set of graphs of size $k$, it is possible to build a system of equations to calculate their frequencies by using the frequencies of the size $k - 1$ graphs and the frequency of a single graph of size $k$. This greatly reduces execution time since the size $k$ graphlet enumeration is much more computationally expensive than the size $k-1$ enumeration followed by solving the system of equations. While this approach is substantially faster than a full enumeration it is quite limited in scope because a new set of equations needs to be manually derived in order to find the $k + 1$ graphlets, $k + 2$, and so on. Similarly to GraphCrunch, Orca manually counts each subgraph, disallowing for different sets of subgraphs to be enumerated. These two tools also do not support edge direction, being only applicable for undirected graphlets.

Network motifs are similar in concept to graphlets and numerous tools for network motif discovery exist. Only the

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCBB.2016.2586046, IEEE/ACM Transactions on Computational Biology and Bioinformatics

census on the original network is performed because we are only concerned with the subgraph enumeration itself and not with assessing motif significance on a large set of randomized networks. Since tools for network motif discovery do not have to the calculate orbit frequencies specific to each node they perform less computational work than graphlet tools. Two well-known methods for the task are Kavosh [47] and Fanmod [48], the latter being an implementation of the ESU algorithm; in our work a more efficient implementation of ESU [33] is used.

Frequent Subgraph Mining (FSM) shares some common aspects with subgraph census: like motif/graphlet algorithms, FSM methods also have to enumerate subgraph occurrences in order to assess which patterns are recurrent; however, FSM algorithms take as input an ensemble of networks instead of a single network. In order to be efficient, FSM tools use pruning strategies to reduce the search space and, therefore, usually do not perform a complete subgraph census. Those strategies make the FSM tools hard to compare with motif discovery algorithms since the latter cannot employ the same optimizations. Since our focus is to evaluate the efficiency of the tools' subgraph enumeration, FSM algorithms are not used for comparison. Nevertheless, the most well-known tools for FSM are gSpan [49], MoFa [6], FFSM [50], gaston [51] and [52].

Because i) GraphCrunch, ii) Orca and iii) ESU/Kavosh methodologies are not directly comparable, three distinct versions of GT-Scanner were used accordingly: i) a version that enumerates all graphlets and orbits of up to size $k$, ii) a version that enumerates up to size $k-1$ graphlets and orbits and then computes a set of equations to calculate the frequencies of the size $k$ graphlets and iii) a version that only enumerates the subgraphs (and not the orbits) of size $k$.

Table 3 presents a detailed comparison between the different tools. In Table 3 (a) it can be observed that, although the networks from Table 2 are relatively small, sometimes more than a 1 billion occurrences are found, showcasing the computational complexity of an exhaustive enumeration and the necessity of an efficient tool. The number of occurrences of $u\mathcal{G}_5$ and $d\mathcal{G}_5$ for the same networks is necessarily the same, however there will probably be many more different directed than undirected graphlets types. Again, this shows the gain in topological information brought by using directed graphlets that is disregarded by undirected graphlets. Additionally, $u\mathcal{G}_5$ takes significantly longer to enumerate than $d\mathcal{G}_4$ since increasing the size of the graphlets greatly increases the computational time. Using GT-Scanner, the time necessary to compute directed graphlets and undirected graphlets of the same size is not substantially different.

Tables 3 (b), (c) and (d) show the results for each different version of GT-Scanner and competing tools. Results showing the average (mean) speedup obtained in all 10 networks for different sets of graphlets, $u\mathcal{G}_5$, $u\mathcal{G}_6$, $d\mathcal{G}_4$ and $d\mathcal{G}_5$, are presented in Table 4. Neither GraphCrunch nor Orca are capable of enumerating either directed graphlets or undirected graphlets with more than 5 nodes. GT-Scanner performs the census faster than the two aforementioned tools and, in addition to that, can enumerate directed graphlets as well as undirected graphlets with more than 5 nodes. Kavosh and ESU can also perform the census for the cases presented here but they are much slower. For instance, for some networks GT-Scanner takes little over a minute to enumerate $u\mathcal{G}_6$ while both Kavosh and ESU need more than a day to complete the same task. On average, GT-Scanner is almost 100 times faster for undirected graphs and about 20 times faster for directed graphs than tools that perform simple subgraph enumeration (Kavosh and ESU). The speedups are lower for directed graphs because the search space is harder to constrain due to the higher number of graphlets sharing less common subtopologies between them. It is noticeable that the speedups of GT-Scanner relative to motif tools are much higher than those relative to graphlet tools. This is due to tools for motif discovery being general, since they can be used to count any set of subgraphs of a given size, while tools to find graphlets can only enumerate undirected graphlets of up to 5 nodes. This allows graphlet tools to have specialized optimizations that motif tools can not match.

From these experiments we can conclude that G-MaGE can both perform faster than state-of-the-art graphlet tools and also provide a more general approach which supports any directed or undirected motif/graphlet size, as long as the set of graphlets fits into memory.

### 3.2.5 Parallel Subgraph Census

Increasing the size of the graphlets from $k$ to $k+1$ makes the enumeration process much more computationally expensive, as can be observed from Table 3. Even a very efficient tool such as GT-Scanner takes a considerable amount of time to compute $u\mathcal{G}_6$ for the metabolic networks, for instance. To deal with this problem we have developed previous work on parallel strategies for subgraph census, applied to both computer clusters [53], [54] and single multicore machines [55], [56]. Past strategies relied on a static division of the work that could not guarantee a balanced division due to highly unbalanced topology of the networks. Additionally, it is impossible to produce a good estimation pre-execution of how the computation should be divided without performing some kind of enumeration beforehand.

A dynamic load-balancing was developed in order to obtain a scalable parallel implementation. The main idea is to divide the work among computers during runtime. The initial work is split evenly by the computing resources and whenever one of them has finished its alloted work it requests more from the other ones (*work-stealing*). Our algorithms perform random polling which has been established as an adequate heuristic for dynamic load balancing [57]. By implementing these ideas, together with an efficient sharing mechanism, we obtained algorithms that scale linearly with the number of processors. This allows for graphlet enumeration to be performed on larger networks or to enumerate bigger graphlets, possibly leading to new insights into the networks. Biologists may have access to computer clusters where they can perform large-scale experiments. Our tool developed for cluster-environments obtained near-linear speedup up to 128 processors, meaning that experiments that took one week to compute would be feasible in less than one hour. On the other hand, our tool for multicore architectures takes advantage of the multiple processors that current PCs have, removing the need to gain access to a specialized cluster.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCBB.2016.2586046, IEEE/ACM Transactions on Computational Biology and Bioinformatics

| $G$ | $\mathcal{G}$ | Occurrences (millions) | Types | GT-Scanner | GraphCrunch | GT-Scanner | ORCA | GT-Scanner | Kavosh | ESU |
|---|---|---|---|---|---|---|---|---|---|---|
| SIG_NCI | $u\mathcal{G}_5$ | 1754 | 17 | 60.73 | 481.45 | 3.07 | 3.08 | 34.08 | 3,524.57 | 2,894.99 |
| | $u\mathcal{G}_6$ | 7883 | 74 | 4,233.55 | n/a | n/a | n/a | 2,051.04 | > 1 day | > 1 day |
| | $d\mathcal{G}_4$ | 24.1 | 89 | 1.70 | n/a | n/a | n/a | 1.13 | 27.20 | 26.16 |
| | $d\mathcal{G}_5$ | 1754 | 842 | 112.09 | n/a | n/a | n/a | 70.26 | 3,669.29 | 3,103.47 |
| SIC_NH | $u\mathcal{G}_5$ | 4.0 | 21 | 1.70 | 6.08 | 0.23 | 0.26 | 0.96 | 56.51 | 45.86 |
| | $u\mathcal{G}_6$ | 1078 | 112 | 44.29 | n/a | n/a | n/a | 24.28 | 1,887.67 | 1,557.45 |
| | $d\mathcal{G}_4$ | 1.55 | 191 | 0.13 | n/a | n/a | n/a | 0.10 | 1.57 | 1.40 |
| | $d\mathcal{G}_5$ | 3.96 | 5219 | 3.71 | n/a | n/a | n/a | 2.61 | 56.16 | 53.79 |
| SIG_SH | $u\mathcal{G}_5$ | 2.55 | 20 | 0.11 | 0.45 | 0.02 | 0.08 | 0.06 | 3.74 | 3.06 |
| | $u\mathcal{G}_6$ | 41.3 | 103 | 2.94 | n/a | n/a | n/a | 1.10 | 76.13 | 62.38 |
| | $d\mathcal{G}_4$ | 0.16 | 79 | 0.01 | n/a | n/a | n/a | 0.01 | 0.18 | 0.16 |
| | $d\mathcal{G}_5$ | 2.55 | 789 | 0.22 | n/a | n/a | n/a | 0.17 | 4.18 | 3.78 |
| SIG_SM | $u\mathcal{G}_5$ | 1.78 | 20 | 0.08 | 0.34 | 0.02 | 0.07 | 0.05 | 2.66 | 2.15 |
| | $u\mathcal{G}_6$ | 26.6 | 102 | 1.30 | n/a | n/a | n/a | 0.72 | 50.00 | 40.38 |
| | $d\mathcal{G}_4$ | 0.12 | 70 | 0.01 | n/a | n/a | n/a | 0.01 | 0.14 | 0.12 |
| | $d\mathcal{G}_5$ | 1.78 | 677 | 0.15 | n/a | n/a | n/a | 0.12 | 3.02 | 2.66 |
| MET_BS | $u\mathcal{G}_5$ | 1455 | 5 | 42.2 | 409.04 | 1.69 | 1.83 | 22.11 | 2,500.87 | 2,436.82 |
| | $u\mathcal{G}_6$ | 1510 | 15 | 3,592.02 | n/a | n/a | n/a | 1,695.38 | > 1 day | > 1 day |
| | $d\mathcal{G}_4$ | 17.1 | 36 | 1.03 | n/a | n/a | n/a | 0.75 | 17.43 | 15.66 |
| | $d\mathcal{G}_5$ | 1455 | 271 | 96.99 | n/a | n/a | n/a | 62.93 | 2,555.38 | 2,334.01 |
| MET_DR | $u\mathcal{G}_5$ | 1782 | 5 | 51.12 | 504.06 | 1.93 | 2.06 | 27.79 | 3,725.94 | 3,002.90 |
| | $u\mathcal{G}_6$ | 2799 | 16 | 4,768.99 | n/a | n/a | n/a | 2,317.72 | > 1 day | > 1 day |
| | $d\mathcal{G}_4$ | 19.5 | 37 | 1.18 | n/a | n/a | n/a | 0.86 | 19.89 | 18.81 |
| | $d\mathcal{G}_5$ | 1.782 | 271 | 117.64 | n/a | n/a | n/a | 79.9 | 3,076.25 | 2,852.23 |
| MET_TY | $u\mathcal{G}_5$ | 1953 | 5 | 56.9 | 551.73 | 2.09 | 2.21 | 30.47 | 4,113.44 | 3,506.87 |
| | $u\mathcal{G}_6$ | 960 | 16 | 5,177.37 | n/a | n/a | n/a | 2,473.07 | > 1 day | > 1 day |
| | $d\mathcal{G}_4$ | 21.3 | 37 | 1.26 | n/a | n/a | n/a | 0.93 | 22.06 | 19.45 |
| | $d\mathcal{G}_5$ | 1953 | 217 | 132.26 | n/a | n/a | n/a | 82.26 | 3,352.5 | 3,141.3 |
| TR_EC | $u\mathcal{G}_5$ | 0.01 | 21 | < 0.01 | 0.03 | < 0.01 | < 0.01 | < 0.01 | 0.02 | 0.01 |
| | $u\mathcal{G}_6$ | 0.04 | 98 | 0.01 | n/a | n/a | n/a | 0.01 | 0.09 | 0.06 |
| | $d\mathcal{G}_4$ | 0.002 | 24 | < 0.01 | n/a | n/a | n/a | < 0.01 | < 0.01 | < 0.01 |
| | $d\mathcal{G}_5$ | 0.01 | 217 | < 0.01 | n/a | n/a | n/a | < 0.01 | 0.02 | 0.01 |
| TR_YST | $u\mathcal{G}_5$ | 2.5 | 20 | 0.09 | 0.73 | 0.02 | 0.07 | 0.05 | 5.00 | 4.04 |
| | $u\mathcal{G}_6$ | 32.0 | 81 | 1.39 | n/a | n/a | n/a | 0.71 | 90.28 | 74.05 |
| | $d\mathcal{G}_4$ | 0.3 | 34 | 0.01 | n/a | n/a | n/a | 0.01 | 0.23 | 0.21 |
| | $d\mathcal{G}_5$ | 2.5 | 174 | 0.18 | n/a | n/a | n/a | 0.14 | 5.13 | 4.74 |

(a) General Information

(b) Graphlet Enumeration: $\{1, 2, ..., k\}$ census

(c) Graphlet Enumeration: $\{1, 2, ..., k$ - $1\}$ census

(d) Subgraph Enumeration: $k$ census

**TABLE 3:** Time comparison (in seconds) of multiple algorithms for graphlet and motif discovery. The three versions of GT-Scanner are compared to algorithms that perform the same tasks: (a) full $\{1, 2, ..., k\}$ enumeration, (b) $\{1, 2, ..., k - 1\}$ enumeration followed by solving a system of equations to obtain the frequencies of the $k$-size graphlets and (c) only enumerate the subgraphs of size $k$ without computing the orbits.

| $\mathcal{G}$ | $\lvert\mathcal{G}\rvert$ | $\lvert\mathcal{O}\rvert$ | GraphCrunch | Orca | Kavosh | ESU |
|---|---|---|---|---|---|---|
| $u\mathcal{G}_5$ | 30 | 73 | $7.15 \pm 2.56$ | $2.04 \pm 1.27$ | $95.00 \pm 30.97$ | $80.11 \pm 27.85$ |
| $u\mathcal{G}_6$ | 142 | 480 | n/a | n/a | $85.89 \pm 24.07$ | $70.31 \pm 19.88$ |
| $d\mathcal{G}_4$ | 214 | 730 | n/a | n/a | $20.61 \pm 3.80$ | $18.73 \pm 3.86$ |
| $d\mathcal{G}_5$ | 9,578 | 45,637 | n/a | n/a | $35.00 \pm 9.77$ | $31.75 \pm 8.30$ |

(a) (b) (c) (d)

**TABLE 4:** Performance comparison between GT-Scanner and other algorithms. *(a)* shows a description of the set of subgraphs being enumerated, as well as the total number of graphlets ($|\mathcal{G}|$) and orbits ($|\mathcal{O}|$). The speedups between our methods and other algorithms are shown in (b), (c) and (d).

## 4 Conclusions

Recent advances in high-throughput cell biology caused enormous amounts of cellular biological data to be continuously produced. Studying the large computational networks resulting from this information can lead to new insight into cellular organization. Due to the size of these networks it is necessary to resort to studying their smaller components such as graphlets network motifs.

Graphlets in particular have been extensively applied to PPIs and other undirected networks but their applicability in directed biological networks, such as cell signaling, transcriptional regulatory and metabolic networks, is limited since they do not consider edge direction. In this paper we have highlighted the importance of adapting graphlets to take into account edge direction by showing that networks of different types (both synthetic and real) can be accurately grouped using directed graphlets.

We have presented an efficient tool, GT-Scanner, that is able to compute directed and undirected graphlets of arbitrary size, as well as network motifs, as long as they fit into memory. GT-Scanner also allows the user to customize the set of graphs that he/she wants to query in the network, further demonstrating the flexibility of our tool. We assess GT-Scanner's performance on a set of directed biological networks and compare it to other tools for graphlet and network motif discovery. We observe that it is the fastest available tool for either task while also being a very general approach. Therefore, we believe that we have broadened the applicability of graphlets by extending them to directed graphlets and by providing an efficient tool for that task.

### Acknowledgements

This work is partially funded by FCT (Portuguese Foundation for Science and Technology) within project UID/EEA/50014/2013. David Aparício is supported by a FCT/MAP-i PhD research grant (PD/BD/105801/2014).

## References

[1] M. Schena, D. Shalon, R. W. Davis, and P. O. Brown, "Quantitative monitoring of gene expression patterns with a complementary dna microarray," *Science*, vol. 270, no. 5235, pp. 467–470, 1995.

[2] A. Wagner and D. A. Fell, "The small world inside large metabolic networks," *Proceedings of the Royal Society of London. Series B: Biological Sciences*, vol. 268, no. 1478, pp. 1803–1810, 2001.

[3] H. Jeong, B. Tombor, R. Albert, Z. N. Oltvai, and A.-L. Barabási, "The large-scale organization of metabolic networks," *Nature*, vol. 407, no. 6804, pp. 651–654, 2000.

[4] O. Sporns, D. R. Chialvo, M. Kaiser, and C. C. Hilgetag, "Organization, development and function of complex brain networks," *Trends in cognitive sciences*, vol. 8, no. 9, pp. 418–425, 2004.

[5] M. P. van den Heuvel, C. J. Stam, R. S. Kahn, and H. E. H. Pol, "Efficiency of functional brain networks and intellectual performance," *The Journal of neuroscience*, vol. 29, no. 23, pp. 7619–7624, 2009.

[6] C. Borgelt and M. R. Berthold, "Mining molecular fragments: Finding relevant substructures of molecules," in *Data Mining, 2002. ICDM 2003. Proceedings. 2002 IEEE International Conference on*. IEEE, 2002, pp. 51–58.

[7] R. Milo, S. Shen-Orr, S. Itzkovitz, N. Kashtan, D. Chklovskii, and U. Alon, "Network motifs: simple building blocks of complex networks." *Science (New York, N.Y.)*, vol. 298, no. 5594, pp. 824–827, 2002.

[8] R. J. Prill, P. A. Iglesias, and A. Levchenko, "Dynamic properties of network motifs contribute to biological network organization," *PLoS biology*, vol. 3, no. 11, p. e343, 2005.

[9] I. Albert and R. Albert, "Conserved network motifs allow protein–protein interaction prediction," *Bioinformatics*, vol. 20, no. 18, pp. 3346–3352, 2004.

[10] S. S. Shen-Orr, R. Milo, S. Mangan, and U. Alon, "Network motifs in the transcriptional regulation network of Escherichia coli," *Nature genetics*, vol. 31, no. 1, pp. 64–68, 2002.

[11] E. R. Shellman, C. F. Burant, and S. Schnell, "Network motifs provide signatures that characterize metabolism," *Molecular BioSystems*, vol. 9, no. 3, pp. 352–360, 2013.

[12] W. Li, L. Chen, X. Li, X. Jia, C. Feng, L. Zhang, W. He, J. Lv, Y. He, W. Li, and Others, "Cancer-related marketing centrality motifs acting as pivot units in the human signaling network and mediating cross-talk between biological pathways," *Molecular BioSystems*, vol. 9, no. 12, pp. 3026–3035, 2013.

[13] 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*, vol. 303, no. 5663, pp. 1538–1542, March 2004.

[14] N. Pržulj, "Biological network comparison using graphlet degree distribution," *Bioinformatics*, vol. 23, pp. 177–183, 2007.

[15] P. Ribeiro, F. Silva, and M. Kaiser, "Strategies for Network Motifs Discovery," in *5th IEEE International Conference on e-Science (e-Science 2009), Oxford, UK*. IEEE Press, December 2009, pp. 80–87.

[16] O. Kuchaiev and N. Pržulj, "Learning the structure of protein-protein interaction networks." in *Pacific Symposium on Biocomputing*, vol. 14, 2009, pp. 39–50.

[17] K. Sun, J. P. Gonçalves, C. Larminie, and Others, "Predicting disease associations via biological network analysis," *BMC bioinformatics*, vol. 15, no. 1, p. 304, 2014.

[18] B. Yoo, H. Chen, F. E. Faisal, and T. Milenković, "Improving identification of key players in aging via network de-noising," in *Proceedings of the 5th ACM Conference on Bioinformatics, Computational Biology, and Health Informatics*. ACM, 2014, pp. 164–173.

[19] O. Kuchaiev, P. T. Wang, Z. Nenadic, and N. Pržulj, "Structure of brain functional networks," in *Engineering in Medicine and Biology Society, 2009. EMBC 2009. Annual International Conference of the IEEE*. IEEE, 2009, pp. 4166–4170.

[20] O. N. Yaveroğlu, T. Milenković, and N. Pržulj, "Proper evaluation of alignment-free network comparison methods," *Bioinformatics*, p. btv170, 2015.

[21] Y. Hulovatyy, H. Chen, and T. Milenkovic, "Exploring the structure and function of temporal networks with dynamic graphlets," *Bioinformatics*, vol. 31, no. 12, pp. i171–i180, 2015.

[22] N. Malod-Dognin and N. Pržulj, "GR-Align: Fast and flexible alignment of protein 3D structures using graphlet degree similarity," *Bioinformatics*, vol. 30, no. 9, pp. 1259–1265, 2014.

[23] D. Garlaschelli and M. I. Loffredo, "Patterns of link reciprocity in directed networks," *Physical Review Letters*, vol. 93, no. 26 I, pp. 1–4, 2004.

[24] S. Mangan and U. Alon, "Structure and function of the feed-forward loop network motif," *Proceedings of the National Academy of Sciences*, vol. 100, no. 21, pp. 11 980–11 985, 2003.

[25] P. Wang, J. Lü, and X. Yu, "Identification of important nodes in directed biological networks: A network motif approach," *PloS one*, vol. 9, no. 8, p. e106132, 2014.

[26] C. Y. Park, D. C. Hess, C. Huttenhower, and O. G. Troyanskaya, "Simultaneous genome-wide inference of physical, genetic, regulatory, and functional pathway components," *PLoS computational biology*, vol. 6, no. 11, p. e1001009, 2010.

[27] S. Wasserman and K. Faust, *Social network analysis: Methods and applications*. Cambridge university press, 1994, vol. 8.

[28] F. Schreiber and H. Schwobbermeyer, "Towards motif detection in networks: frequency concepts and flexible search," *Proc. Intl. Wsh. Network Tools and Applications in Biology (NETTABâĂŹ04)*, pp. 91–102, 2004.

[29] P. Ribeiro and F. Silva, "G-tries: an efficient data structure for discovering network motifs," in *Proceedings of the 2010 ACM Symposium on Applied Computing*. ACM, 2010, pp. 1559–1566.

[30] T. Milenkoviæ and N. Pržulj, "Uncovering biological network function via graphlet degree signatures," *Cancer informatics*, vol. 6, p. 257, 2008.

[31] P. Paredes and P. Ribeiro, "Towards a faster network-centric subgraph census," in *Advances in Social Networks Analysis and Mining (ASONAM), 2013 IEEE/ACM International Conference on*. IEEE, 2013, pp. 264–271.

[32] P. Ribeiro and F. Silva, "G-Tries: A data structure for storing and finding subgraphs," *Data Mining and Knowledge Discovery*, vol. 28, no. 2, pp. 337–377, 2014.

[33] P. Ribeiro, "Efficient and scalable algorithms for network motifs discovery," Ph.D. dissertation, Faculty of Science of the University of Porto, June 2011.

[34] S. Choobdar, P. Ribeiro, and F. Silva, "Motif mining in weighted networks," in *Data Mining Workshops (ICDMW), 2012 IEEE 12th International Conference on*. IEEE, 2012, pp. 210–217.

[35] P. Ribeiro and F. Silva, "Discovering Colored Network Motifs," in *Complex Networks V*. Springer, 2014, pp. 107–118.

[36] P. Erdös and A. Rényi, "{On the evolution of random graphs}," *Publ. Math. Inst. Hung. Acad. Sci*, vol. 5, pp. 17–61, 1960.

[37] K.-I. Goh, B. Kahng, and D. Kim, "Universal behavior of load distribution in scale-free networks," *Physical Review Letters*, vol. 87, no. 27, p. 278701, 2001.

[38] J. Leskovec, J. Kleinberg, and C. Faloutsos, "Graphs over time: densification laws, shrinking diameters and possible explanations," in *Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*. ACM, 2005, pp. 177–187.

[39] T. F. Cox and M. A. Cox, *Multidimensional scaling*. CRC Press, 2000.

[40] Ö. N. Yaveroğlu, N. Malod-Dognin, D. Davis, Z. Levnajic, V. Janjic, R. Karapandza, A. Stojmirovic, and N. Pržulj, "Revealing the hidden language of complex networks," *Scientific reports*, vol. 4, 2014.

[41] C. F. Schaefer, K. Anthony, S. Krupa, J. Buchoff, M. Day, T. Hannay, and K. H. Buetow, "Pid: the pathway interaction database," *Nucleic acids research*, vol. 37, no. suppl 1, pp. D674–D679, 2009.

[42] Q. Cui, Y. Ma, M. Jaramillo, H. Bari, A. Awan, S. Yang, S. Zhang, L. Liu, M. Lu, M. O'Connor-McCourt, and Others, "A map of human cancer signaling," *Molecular systems biology*, vol. 3, no. 1, p. 152, 2007.

[43] A. Ma'ayan, S. L. Jenkins, R. L. Webb, S. I. Berger, S. P. Purushothaman, N. S. Abul-Husn, J. M. Posner, T. Flores, and R. Iyengar, "Snavi: Desktop application for analysis and visualization of large-scale signaling networks," *BMC systems biology*, vol. 3, no. 1, p. 10, 2009.

[44] C. J. Honey, J.-P. Thivierge, and O. Sporns, "Can structure predict function in the human brain?" *Neuroimage*, vol. 52, no. 3, pp. 766–776, 2010.

[45] T. Milenković, J. Lai, and N. Pržulj, "Graphcrunch: a tool for large network analyses," *BMC bioinformatics*, vol. 9, no. 1, p. 70, 2008.

[46] T. Hočevar and J. Demšar, "A combinatorial approach to graphlet counting," *Bioinformatics*, vol. 30, no. 4, pp. 559–565, 2014.

[47] Z. R. Kashani, H. Ahrabian, E. Elahi, A. Nowzari-Dalini, E. S. Ansari, S. Asadi, S. Mohammadi, F. Schreiber, and A. Masoudi-Nejad, "Kavosh: a new algorithm for finding network motifs," *BMC bioinformatics*, vol. 10, no. 1, p. 318, 2009.

[48] S. Wernicke and F. Rasche, "Fanmod: a tool for fast network motif detection," *Bioinformatics*, vol. 22, no. 9, pp. 1152–1153, 2006.

[49] X. Yan and J. Han, "gspan: Graph-based substructure pattern mining," in *Data Mining, 2002. ICDM 2003. Proceedings. 2002 IEEE International Conference on*. IEEE, 2002, pp. 721–724.

[50] J. Huan, W. Wang, and J. Prins, "Efficient mining of frequent subgraphs in the presence of isomorphism," in *Data Mining, 2003. ICDM 2003. Third IEEE International Conference on*. IEEE, 2003, pp. 549–552.

[51] S. Nijssen and J. N. Kok, "The gaston tool for frequent subgraph mining," *Electronic Notes in Theoretical Computer Science*, vol. 127, no. 1, pp. 77–87, 2005.

[52] N. Jin, C. Young, and W. Wang, "Gaia: graph classification using evolutionary computation," in *Proceedings of the 2010 ACM SIGMOD International Conference on Management of data*. ACM, 2010, pp. 879–890.

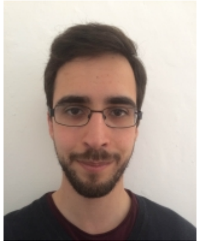
[53] P. Ribeiro, F. Silva, and L. Lopes, "Efficient parallel subgraph counting using g-tries," in *Cluster Computing (CLUSTER), 2010 IEEE International Conference on*. IEEE, 2010, pp. 217–226.

[54] ——, "Parallel discovery of network motifs," *Journal of Parallel and Distributed Computing*, vol. 72, no. 2, pp. 144–154, 2012.

[55] D. Aparício, P. Ribeiro, and F. Silva, "Parallel subgraph counting for multicore architectures," in *Parallel and Distributed Processing with Applications (ISPA), 2014 IEEE International Symposium on*. IEEE, 2014, pp. 34–41.

[56] D. Aparício, P. Paredes, and P. Ribeiro, "A scalable parallel approach for subgraph census computation," in *Euro-Par 2014: Parallel Processing Workshops*. Springer, 2014, pp. 194–205.

[57] P. Sanders, "A detailed analysis of random polling dynamic load balancing," in *Parallel Architectures, Algorithms and Networks, 1994.(ISPAN), International Symposium on*. IEEE, 1994, pp. 382–389.



**David Aparício** a Ph. D. student in Computer Science from the MAP-i Doctoral Program, which is offered jointly by the Universities of Minho, Aveiro and Porto, in Portugal. He received his M. Sc. in Computer Science from the University of Porto (UP) in 2014. He is also a researcher in CRACS & INESC-TEC since 2012. His primary research interests are in complex network analysis, graph data mining, parallel/distributed computing and bioinformatics.

**Pedro Ribeiro** is an Assistant Professor in the Computer Science Department of the School of Sciences at the University of Porto, Portugal. He was awarded his Ph.D. in Computer Science from the University of Porto in 2011. He has a core background on algorithms and data structures and his main research line is focused on network science and on the analysis of complex networks, particularly on the algorithmic aspects of discovering interesting patterns. He also has a strong interest in parallel and distributed computing, computer science education and programming contests.

**Fernando Silva** is a Full Professor in the Computer Science Department of the School of Sciences at the University of Porto (UP), Portugal. He was awarded his Ph.D. in Computer Science from the University of Manchester, UK (1993), and, in 2007, obtained the Habilitation in Informatics from the New University of Lisbon, Portugal. He currently coordinates the Center for Research in Advanced Computing Systems (CRACS) of INESC TEC, and is a member of the scientific board of MAP-i, the Doctoral Program in Computer Science of the Universities of Minho, Aveiro and Porto (was director in 2008/09 and 2014/15 editions). His primary research interests are in programming languages, parallel and distributed computing, mobile edge computing, algorithms and applications in information mining. He has advised 10 completed PhD theses, coordinated research projects totalling over 5 million Euros, and published extensively in these areas.