Identification of co-existing embeddings of a motif in multilayer networks

Yuanfang Ren
yuanfang@cise.ufl.edu
University of Florida
Gainesville, FL, USA

Aisharjya Sarkar
sarkar@cise.ufl.edu
University of Florida
Gainesville, FL, USA

Aysegul Bumin
aysegul.bumin@ufl.edu
University of Florida
Gainesville, FL, USA

Kejun Huang
kejun.huang@ufl.edu
University of Florida
Gainesville, FL, USA

Pierangelo Veltri
veltri@unicz.it
Magna Graecia University
Catanzaro, Italy

Alin Dobra
adobra@cise.ufl.edu
University of Florida
Gainesville, FL, USA

Tamer Kahveci
tamer@cise.ufl.edu
University of Florida
Gainesville, FL, USA

ABSTRACT

Interactions among molecules, also known as biological networks, are often modeled as binary graphs, where nodes and edges represent the molecules and the interaction among those molecules, such as signal transmission, genes-regulation, and protein-protein interactions. Subgraph patterns which are recurring in these networks, called motifs, describe conserved biological functions. Although traditional binary graph provides a simple model to study biological interactions, it lacks the expressive power to provide a holistic view of cell behavior as the interaction topology alters and adopts under different stress conditions as well as genetic variations. Multilayer network model captures the complexity of cell functions for such systems. Unlike the classic binary network model, multilayer network model provides an opportunity to identify conserved functions in cell among varying conditions. In this paper, we introduce the problem of co-existing motifs in multilayer networks. These motifs describe the dual conservation of the functions of cells within a network layer (i.e., cell condition) as well as across different layers of networks. We propose a new algorithm to solve the co-existing motif identification problem efficiently and accurately. Our experiments on both synthetic and real datasets demonstrate that our method identifies all co-existing motifs at near 100 % accuracy for all networks we tested on, while competing method's accuracy varies greatly between 10 to 95 %. Furthermore, our method runs at least an order of magnitude faster than state of the art motif identification methods for binary network models.

BCB '22, August 7–10, 2022, Northbrook, IL, USA

ACM ISBN 978-1-4503-9386-7/22/08…$15.00
https://doi.org/10.1145/3535508.3545528

KEYWORDS

co-existing motifs, motifs, multilayer networks

ACM Reference Format:
Yuanfang Ren, Aisharjya Sarkar, Aysegul Bumin, Kejun Huang, Pierangelo Veltri, Alin Dobra, and Tamer Kahveci. 2022. Identification of co-existing embeddings of a motif in multilayer networks. In *13th ACM International Conference on Bioinformatics, Computational Biology and Health Informatics (BCB '22), August 7–10, 2022, Northbrook, IL, USA.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3535508.3545528

1 INTRODUCTION

Biological networks describe a system of interacting molecules. Through these interactions, these molecules carry out key functions such as regulation of transcription [24] and transferring signals [40]. Analyzing the topologies of the biological networks that govern key cellular functions has already provided important insights into these functions [6, 19, 29, 39, 47]. Existing methods often model biological networks as binary graphs, with nodes and edges representing interacting molecules (e.g., proteins or genes) and the interactions between them respectively [8]. They typically represent a graph by a tuple $G = (V, E)$, where V is the set of nodes and $E \subseteq V \times V$ is the set of edges that connect pairs of nodes [11]. Such traditional binary graph models have been beneficial in studying cellular processes under specific conditions. They, however, have been inadequate in providing a holistic view of cells as the interactions between molecules can take place in various forms. For instance, a gene can regulate the transcription of another gene through its promoter region, while interacting through their products in a metabolic reaction or a signaling event, leading to multiple interactions among the same set of molecules under different dynamics. The interaction patterns between molecules can be further altered through various factors, such as genetic or epigenetic mutations, variations in DNA replication, and environmental factors (e.g., oxidative stress) [25]. We model such complex relationships between molecules with an extended graph model, named *multilayer network*, where each layer of the network describes the set interactions under one condition.

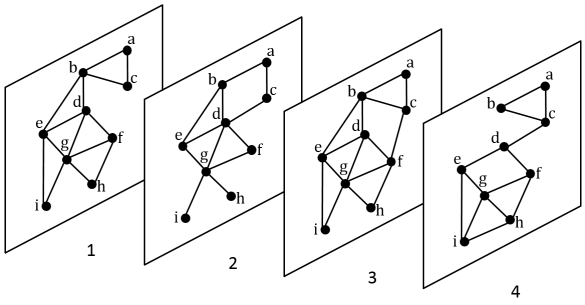


Figure 1: A multilayer network $\mathcal{G}$ with four layers and nine nodes.

Figure 1 presents a multilayer network with four layers and nine nodes.

Studying multilayer networks has great potential to provide new insights into systems biology as they describe alternative interaction patterns collectively. Several existing studies focus on mathematical modeling of these networks [20, 33]. Some others adapt existing measures for characterizing binary networks to multilayer networks, such as centrality [1, 42], clustering coefficient [28], correlations [34]. Another class of studies computes various dynamical processes on multilayer topologies, such as percolation [4], diffusion processes [12], cooperation [13], synchronization [35], and information spreading [37]. An extensive review of the studies on multilayer networks can be found in [5, 21].

Network motifs are patterns of local interconnections occurring significantly more (or less) in a given network than in a random network of the same size [31]. There are several definitions of motif counts in the literature. The naive motif count simply counts all instances of a given motif topology in the given network. Notice that, in this definition different motif instances can share the same node or edge on the given graph. More restricted definitions of motif counts limit or prevent such overlaps to ensure that different instances can be realized simultaneously. For both the naive and restricted motif count definitions, network motif discovery is a computationally hard problem as it requires solving the well-known subgraph isomorphism problem, which is NP-complete [10].

Motif topologies characterize the structure of networks and explain highly conserved functions saved through interactions among molecules. Motifs have been successfully used in many applications, such as studying the biological processes that regulate transcription [41], finding important genes that affect the spread of infectious diseases [44] and revealing relationships across species [14, 16].

Although motifs are studied in the context of stand alone (i.e., single layer and static) networks and dynamic networks (e.g., [9, 15, 17, 22, 23, 36, 45]), little research has been devoted to studying motifs in multilayer networks. Furthermore, there is no generalized definition of motifs in multilayer networks. In the literature [2, 3, 27], the closest problem to multilayered network motifs arises in the concept of multi-link networks, which is the organization of edges between the same pair of nodes across all layers. Such multi-link motifs specify each layer's organization of edges among a set of nodes. However, the number of such distinct multi-link motifs grows exponentially with the number of nodes and layers. Thus, all these studies investigate motifs that contain a limited number of layers and a small set of nodes. Thus, this problem remains understudied in-depth for multilayer networks.

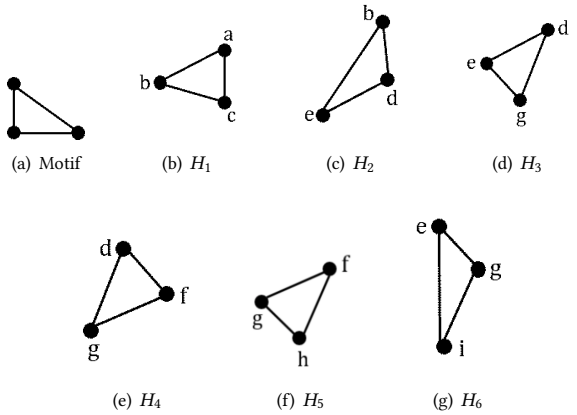
Contributions. In this paper, we introduce new formulations of motifs. Unlike existing motif concepts in the literature, our formulation captures the dependencies between different layers of a multilayer network. We call these motifs *co-existing motifs*. Briefly, we say that a set of placements of a given motif topology is *co-existing* if the absence/presence of each placement depends on that of the remaining ones. This definition allows us to identify conserved as well as unique patterns of functions of biological systems under different conditions. We propose a novel algorithm for counting the maximal set of independent embeddings of a given motif $\mathcal{M}$ that co-exist in a given multilayer network $\mathcal{G}$. Our method works in four steps. In the first step, we construct an aggregate graph corresponding to $\mathcal{G}$. The aggregate graph transforms the problem from multilayer to single layer without any loss of information. We then construct a filtered graph from an aggregate graph consisting of only the edges that fulfill the minimum layer support. In the second step, we identify all possible embeddings in $\mathcal{G}$ using a filtered graph and select the potential candidate embeddings. In the third step, we classify subsets of embeddings identified in the previous step based on the layers in which they coappear and layer support such that the count of layers in which the embeddings exist is greater than the minimum layer support. Finally, we use a heuristic strategy to identify the maximal set of independent embeddings that co-exist across different layers. For that, we first build an overlap graph based on the $\mathcal{M}$, where each node corresponds to an embedding. Then we calculate the loss value for each node in the overlap graph and iteratively pick the node with the least loss. It includes the corresponding embedding to the result set and removes this node along with all nodes that conflict with this selection from the overlap graph. We repeat this process until the overlap graph is empty. Our experiments on both synthetic and real datasets demonstrate that our method identifies all co-existing motifs at near 100 % accuracy for all networks we tested on, while competing method's accuracy varies greatly between 10 to 95 %. Furthermore, our method runs at least an order of magnitude faster than state of the art motif identification methods for binary network models.

The rest of the paper is organized as follows. We define our model for co-existing motifs in multilayer networks and our algorithm to identify these motifs (Section 2). We evaluate our method experimentally and we present results and discussion in Section 3; finally, in Section 4 we conclude with a brief discussion. [1]

2 METHODS

Preliminaries and notation. A multilayer network is a set of k networks with $k > 1$, such that each network consists of the same set of nodes, but possibly a different set of edges. Formally a multilayer network is a (k+2)-tuple $\mathcal{G} = (V, E_1, E_2, \ldots, E_k, L)$, where V is a set of nodes, each E_i denotes the set of edges in the ith layer of this network, and $L = \{1, 2, \ldots, k\}$ denotes the unique labels of each of the k layers. Figure 1 shows a multi-layer

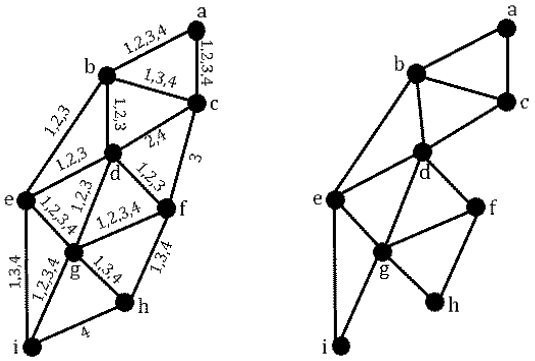
[1]This paper is partially funded by NSF under Award Number 2111679.



(a) Motif (b) H_1 (c) H_2 (d) H_3

(e) H_4 (f) H_5 (g) H_6

Figure 2: (a) A triangle motif; (b) An embedding H_1 that exists in layers $\{1, 3, 4\}$ of Figure 1; (c) An embedding H_2 that exists in layers $\{1, 2, 3\}$; (d) An embedding H_3 that exists in layers $\{1, 2, 3\}$; (e) An embedding H_4 that exists in layers $\{1, 2, 3\}$; (f) An embedding H_5 that exists in layers $\{1, 3, 4\}$. (g) An embedding H_6 that exists in layers $\{1, 3, 4\}$.



(a) Aggregate graph $\mathcal{A}$ (b) Filtered graph $\mathcal{F}$

Figure 3: (a) The aggregate graph $\mathcal{A}$ of the multi-layer network $\mathcal{G}$ in Figure 1. The labels associated with each edge represents the layers containing it. (b) The filtered graph $\mathcal{F}$ containing edges occurring in at least a given number of layers (here, the cutoff for the number of layers is 2). The edges (c, f) and (h, i) are removed in $\mathcal{F}$.

network consisting of nine nodes ($V = \{a, b, \ldots, i\}$) and four layers ($L = \{1, 2, 3, 4\}$).

We define a *motif pattern* as a connected graph $M = (V', E')$, where V' and E' denote the set of motif nodes and edges respectively. Figure 2(a) represents an example of motif pattern.

Consider a multilayer network $\mathcal{G} = (V, E_1, E_2, \ldots, E_k, L)$. For each $i \in \{1, 2, \ldots, k\}$, we denote the network at the ith layer of $\mathcal{G}$ with $G_i = (V, E_i)$. Given a motif topology M, we say that a subset of edges of G_i, denoted by E_i' is an *embedding* of M if the subgraph

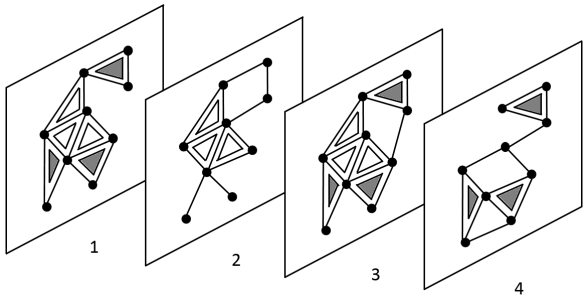


Figure 4: The potential group of embeddings of triangle pattern that co-exist in the multi-layer network $\mathcal{G}$ in Figure 1. There are six embeddings that persists across four layer(s). Each embedding is marked with an inner triangle (shaded/un-shaded). The shaded group of three embeddings co-exist in layers $\{1, 3, 4\}$ and the un-shaded group of three embeddings co-exist in layers $\{1, 2, 3\}$.

of G_i induced by the edges E_i' is isomorphic to M. For instance, Consider layer 1 of the multilayer network in Figure 1, and the motif pattern in Figure 2(a). Let us denote the network at this layer with G_1. There are six possible embeddings of this motif in G_1, shown by $H_1, H_2, \ldots, H_6$ in Figures 2(b) to 2(g). We will denote the existence of each embedding H_i using the set containment symbol "$\in$" (such as $H_1 \in G_1$ means that H_1 is an embedding of the given motif in G_1). We say that two embeddings of M *overlap* if they share at least one edge. For instance, in Figure 2, among the six embeddings $H_1, H_2, \ldots, H_6$, the two embedding H_1 and H_2 do not overlap. On the other hand, H_2 and H_3 overlap since they have the common edge (e, d).

Problem definition. Given $\mathcal{G}$, a motif topology $\mathcal{M}$, and minimum appearance frequency $f \in [1{:}k]$, the goal is to find the largest set of non-overlapping embeddings of $\mathcal{M}$, $\mathcal{H} = \{H_1, H_2, \ldots, H_m\}$, such that the following two conditions hold:

(i) For all G_i, either $\forall H \in \mathcal{H}$, $H \in G_i$ or $\forall H \in \mathcal{H}$, $H \notin G_i$.
(ii) For all $H \in \mathcal{H}$, H is in at least f layers of $\mathcal{G}$.

Let us define an indicator function $\psi(\cdot, \cdot)$ which takes a network layer G_i and a motif embedding H as input as,

$$\psi(G_i, H) = \begin{cases} 1, & \text{if } H \in G_i \\ 0, & \text{otherwise} \end{cases}$$

We can write (i) above as follows:

$$\prod_{G_i} \Big(\prod_{H \in \mathcal{H}} \psi(G_i, H) + \prod_{H \in \mathcal{H}} (1 - \psi(G_i, H)) \Big) = 1 \tag{1}$$

Similarly, we express condition (ii) above as:

$$\sum_{i} \prod_{H \in \mathcal{H}} \psi(G_i, H) \geq f \tag{2}$$

Proposed solution. Our algorithm takes a multilayer network $\mathcal{G} = (V, E_1, E_2, \ldots, E_k, L)$, a motif pattern $\mathcal{M}$ and an integer f denoting minimum number of layers containing an embedding. Our algorithm has four major steps:

Step 1. Construct auxiliary networks. We construct a summary of a given multilayer network $\mathcal{G}$, and call it the *aggregate network*.

This network aggregates all the layers of $\mathcal{G}$ into a single layer. We denote the aggregate network with $\mathcal{A}_{\mathcal{G}} = (V, \mathcal{E}, \Omega)$. Here, V is the same set of nodes as that of $\mathcal{G}$. $\mathcal{E}$ is the set of all the edges that appears in at least one layer of $\mathcal{G}$, that is $\mathcal{E} = \cup_{i=1}^{k} E_i$. The function $\Omega : \mathcal{E} \rightarrow \{L \cup \{\emptyset\}\}^k$ returns the set of layers which contain a given edge in $\mathcal{G}$. That is, for each $(u, v) \in \mathcal{E}$, $\Omega(u, v) = \{\ell | (u, v) \in E_\ell\}$. Figure 3(a) shows the aggregate network of the multilayer network in Figure 1. In this figure, edge (a, b) is labeled as 1, 2, 3, 4 since (a, b) appears in all four layers. Edge (b, c) is labeled with 1, 3, 4, as this edge appears in layers 1, 3, and 4. Hence $\Omega(a, b) = \{1, 2, 3, 4\}$, and $\Omega(b, c) = \{1, 3, 4\}$.

Notice that as the number of layers k in the given multilayer networks grow, and as the topologies of interactions at different layers deviate, the aggregate network gets dense. We construct a sparser network, called *filtered network* $\mathcal{F}_{\mathcal{G},f} = (V, \mathcal{E}_f)$ by removing those edges from $\mathcal{A}$ which are present in less than f number of layers in $\mathcal{G}$. Formally, $\mathcal{E}_f = \{e \in \mathcal{E} | f \leq |\Omega(e)|\}$. For example, for $f = 2$, the aggregate network in Figure 3(a) yields the filtered network in Figure 3(b), as we remove the two edges (c, f) and (h, i) which have layer support less than 2.

Step 2. Identify candidate embeddings. Given a motif $\mathcal{M}$, we identify the set of all possible embeddings $\mathcal{H}^0$ using the aggregate and filtered networks $\mathcal{A}$ and $\mathcal{F}$. Consider a subnetwork H of $\mathcal{F}$ such that H is isomorphic to the given motif $\mathcal{M}$. We say that H belongs to $\mathcal{H}^0$ if H is a subnetwork of at least f layers of the multilayer network $\mathcal{G}$. Mathematically, we express this constraint as, $|\cap_{(u,v)\in H} \Omega(u, v)| \geq f$. For example, for the triangle motif pattern in Figure 2(a), Figures 2(b) to 2(g) show all six possible candidate embeddings of this motif in the filtered network in Figure 3(b) which also have a minimum support from $f = 2$ network layers. Notice that, in $\mathcal{F}$ although there are patterns isomorphic to the $\mathcal{M}$ (such as the set of edges $(b, c), (c, d), (b, d)$), they are not included in the candidate set $\mathcal{H}^0$, since the size of the intersection of the three layer sets corresponding to those edges is less than f.

Step 3. Classify embeddings based on supporting layers. We partition all motif embeddings found in the previous step into classes, where each class is identified as the set of layers $\mathcal{L} \subseteq L$ containing the embeddings in that class. In this manner, we get a set of classes C, where each member $c_i \in C$ represents a tuple of sets $(\mathcal{L}_i, \mathcal{H}_{c_i})$, where $\mathcal{L}_i \subseteq L$ and $\mathcal{H}_{c_i} \subseteq \mathcal{H}^0$. For example, the embeddings H_1, H_5 and H_6 in Figure 2 exists in layers $\{1, 3, 4\}$ of the multilayer network in Figure 1. Similarly, the embeddings H_2, H_3, H_4 exists in layers $\{1, 2, 3\}$. Therefore, $C = \{c_1 = (\{1, 3, 4\}, \{H_1, H_5, H_6\}), c_2 = (\{1, 2, 3\}, \{H_2, H_3, H_4\})\}$ consists of two such classifications. Figure 4 shows the two classes of this motif topology as shaded and white triangles.

Step 4. Identify nonoverlapping embeddings. This step consists of the following two sub-steps:

(a) *Construct overlap graph.* Once we identify all candidate embeddings using $\mathcal{H}^0$, we build a graph which models the overlap pattern among these candidate embeddings. We call this the *overlap graph* and denote it with O. Each node of O corresponds to an embedding $H_i \in O$. We insert an edge between two nodes if their corresponding embeddings share at least an edge. Figure 5 shows the overlap graph of the filtered graph $\mathcal{F}$ in Figure 3(b) of the multilayer network in Figure 1. For example, the embeddings H_2 and H_3 overlap for they share edge (d, e). Thus, we connect the nodes corresponding to H_2 and H_3 by an edge in Figure 5. Similarly, H_3 shares its edge (d, e) with H_2, (d, g) with H_4, and (e, g) with H_6. Thus, we connect the node for H_3 to those for H_2, H_4, and H_6. On the other hand, H_1 does not share an edge with any other embedding. Therefore, the node corresponding to H_1 is isolated.

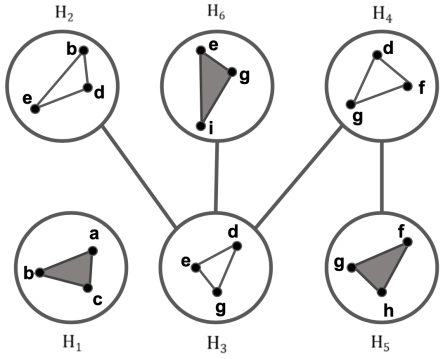
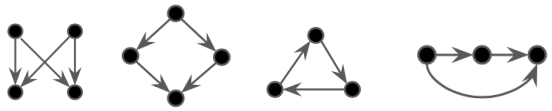


Figure 5: Overlap graph for the embeddings $H_1, H_2, H_3, H_4, H_5, H_6$. The embeddings H_1, H_5, H_6 co-exist in layers $\{1, 3, 4\}$. The embeddings H_2, H_3, H_4 co-exist in layers $\{1, 2, 3\}$.

(b) *Select nonoverlapping embeddings per classification.* For each class $c_i \in C$, we build an overlap graph O_{c_i} which is the induced subgraph of O. For each O_{c_i}, we iteratively select nodes as follows. At each iteration, we first select the node (corresponding embedding H_j) with the smallest degree in O_{c_i}. We then remove that node and all the nodes connected to it, as the nodes connected to it correspond to embeddings which overlap with H_j. We repeat these iterations until the overlap graph O_{c_i} has no nodes left. This yields a set of nonoverlapping and co-existing embeddings which appear in at least f layers denoted by class c_i. After doing this for all classes, we report the maximal set obtained across all classes. In Figure 5 the nodes labeled with H_1, H_5, H_6 correspond to class c_1, and the nodes labeled with H_2, H_3, H_4 correspond to class c_2. The induced subgraph O_{c_1} of the overlap graph contains three disconnected nodes. Thus the three embeddings H_1, H_5, H_6 are non-overlapping, co-existing, and exist in at least f layers. The induced subgraph O_{c_2} has three nodes as well. However, upon choosing the node with lowest degree (such as the node labeled with H_2), node H_3 gets removed as it is connected to H_2. As a result only two embeddings H_2 and H_4 are from this class. In summary class c_1 yields the largest number of non-overlapping, co-existing motif embeddings.

3 RESULTS AND DISCUSSION

In this section, we evaluate the performance of our method. We perform our experiments using both synthetic and real datasets.



(a) Bifan (b) Biparallel (c) Cascade (d) Feed forward

Figure 6: Four conserved motifs studied frequently in the literature.

We consider four motif topologies, namely bifan, biparallel, cascade and feed forward loop (see Figure 6). These four motifs have been extensively studied in the literature and have been shown to be over-represented in many biological networks [31]. We compare our method with a baseline algorithm (explain in detail later). In the following, we describe the datasets, quality measure and implementation details.

Synthetic dataset. To observe the performance of our method under controlled dataset characteristics, We perform experiments on synthetically generated directed multilayer networks. To guarantee that each network contains a set of independent embeddings with coexistence, we plant a set of co-existing embeddings of a given motif topology while we generate these networks. To better describe our synthetic datasets, we first define the necessary terminology. We refer to the number of nodes and the number of edges per node in each layer of the multi-layer network as the *size* ($|V|$) and the *average degree* (d) of the network. We use the term *edge ratio*, denoted with $c \in (0, 1]$, to represent the ratio of the number of edges involved in planted motif embeddings to the total number of edges in the network. Given the network size, average degree, edge ratio and motif pattern having k edges, the number of motif embeddings in each layer of a multi-layer network is thus $|V|d\frac{c}{2k}$. In our experiments, we fix the edge ratio to 0.2. We construct our synthetic networks by varying four parameters: network size, average degree, minimum appearance frequency f and motif type. For these parameters, we use the following values $|V| \in \{200, 400, 800, 1600\}$, $d \in \{2, 4, 6, 8, 10\}$, $f \in \{4, 6, 8, 10\}$, and the four motif types shown in Figure 6. For each parameter setting, we construct a multilayer networks each having 10 layers as follows. We first construct one network layer having $|V|d\frac{c}{2k}$ independent embeddings and randomly generate the remaining edges. We then generate the remaining nine layers by applying topological perturbations on it. We do this using the degree preserving edge shuffling method [30] with a given mutation rate of $\beta \in [0, 1]$. Given the edge set of a network denoted with E', a mutation rate of β means that $\beta \times \frac{|E|'}{2}$ edge pairs in the network are shuffled. In our experiment, we fix $\beta = 0.5$. Given the parameter f, in addition to the first layer containing embedding set, for each of the remaining $f - 1$ layers, we perform perturbation on the edges which do not contain planted embeddings and keep the edges of planted embeddings. Thus, we obtain f layers with each containing the same set of independent motif embeddings but also having some topological difference. As for the remaining $10 - f$ layers, we construct them by performing perturbation on the entire edge set of the first layer. Notice that here we also make sure none of the planted embeddings appear in any of these $10-f$ layers. As a result, we guarantee that there exists a set of embeddings co-existing in the multilayer network with the size greater or equal to the number of planted embeddings. After generating all layers, we randomly placed them to form a multilayer network. We repeat the procedure above to create 10 synthetic multilayer networks. Thus in total we have $4 \times 5 \times 4 \times 4 \times 10 = 1600$ synthetic multilayer networks.

Real Datasets. We perform our algorithm on three datasets based on three different organisms: *Homo sapiens* (human), *Saccharomyces cerevisiae* (yeast) and *Escherichia coli* (E. coli). For each organism, we obtain datasets collected from two sources. The first one is the underlying transcription factor regulatory network corresponding to each organism. The second one is the dataset containing gene expression values under multiple conditions. The reason behind choosing such a dataset having multiple conditions is that we treat each condition as a layer within a multi-layer network. Next we describe each dataset in detail.

- *Homo sapiens.* We extract core human regulatory network consisting of connections among 475 sequence-specific transcription factors across 41 diverse cell and tissue types from [32]. We combine all the interactions present across 41 cell lines. It consists of a total of 38,393 unique, directed regulatory interactions (edges) for all cell lines. The network provides a detailed collection of the circuitry, dynamics, and organizing principles and forms the underlying human transcription factor regulatory network. Next, we extract the dataset GSE62932 [7] consisting of gene expression values from 68 samples at different stages of colorectal cancer. The stages are healthy/control group, stage I, stage II, stage III and stage IV. The samples include four healthy control patient tissues, 12 stage I, 17 stage II, 20 stage III and 15 stage IV colorectal cancer patient tissues. The gene expression values across all stages in the dataset ranges approximately between 2.0 and 12.0. We choose a cutoff value $\sigma = 9.0$ on the gene expression values to construct a multi-layer network. Given a σ, we filter out all the genes that have gene expression values greater or equal to σ for majority of the samples within each stage. In this manner, we get a list of genes for each stage. Next, to construct the network for each layer out of this gene list for a particular stage, we consider all the interactions within the underlying human transcription factor regulatory network where both the source and destination genes are present in the gene list. In this manner, we form all the layers of a multi-layer network.
- *Saccharomyces cerevisiae.* We extract yeast transcription regulatory network from YEASTRACT database [43] which consists of TF-gene regulatory pairs under nine experimental conditions having strong evidence support in the literature that the TF binds to the promoter region of the target gene and the perturbation of the TF affects the target gene's expression significantly [46]. The nine different experimental conditions classified by YEASTRACT database are (1) cycle and morphology, (2) stress, (3) oxygen availability, (4) unstressed log-phase growth (control), (5) nitrogen source quality and availability, (6) carbon source quality and availability, (7) ion, metal, phosphate, sulfur, vitamin availability, (8) lipid supplementation and (9) complex industrial media. Among these nine experimental conditions, we choose unstressed

log-phase growth (control) TF-gene regulatory pairs as the underlying yeast transcription factor regulatory network for our experiments. It consists of a total of 12,219 directed regulatory interactions. Next, we extract the dataset GSE8536 [26] consisting of gene expression values from 21 samples that measures yeast's response throughout a 15 day wine fermentation. It is based on expression measurements of 0.5, 2, 3.5, 7, and 10% ethanol at roughly after 1, 12, 24, 48, 60, 120, and 340 hours. Thus, the dataset has measurements for seven time points, each time point consists of 3 samples. The gene expression values across all time points in the dataset ranges approximately between 4.0 and 8.0. Here, we model each time point (hour) as a layer in a multi-layer network. We choose a cutoff value $\sigma = 6.0$ on the gene expression values to construct a multi-layer network. Given a σ, we filter out all the genes that have gene expression values greater or equal to σ for at least two samples (out of 3) within each time point. In this manner, we get a list of genes for each time point. Next, to construct the network for each layer out of this gene list for a particular time point, we consider all the interactions within the underlying yeast transcription factor regulatory network where both the source and destination genes are present in the gene list. In this manner, we form all the layers of a multi-layer network.

- *Escherichia coli.* We use E.coli transcription regulatory network downloaded from RegulonDB Database [38]. This network contains 4400 nodes and 4407 edges. We use the E.coli gene expression dataset, GSE20305, obtained from the GEO database to determine the existence of each interaction under different time points and different conditions [18]. It contains five different stress conditions including cold, heat, oxidative, lactose diauxie and stationary phase (control). For each network layer, we include an edge in the network at that layer if the gene expression of the reactant gene is greater than the cutoff 8.0.

Competing method. To the best of our knowledge, this is the first study to count co-existing independent embeddings in multi-layer networks. To better evaluate the performance of our method, we develop a baseline method which only depends on identifying embeddings in classic single layer networks as follows. Given a multilayer network with k layers and the parameter f, we first choose a subset of α ($f \leq \alpha \leq k$) layers from all layers on which we aim to find embeddings appearing on these layers while not appearing on the remaining $k-\alpha$ layers. In total, there are $\sum_{\alpha=f}^{k} \binom{k}{\alpha}$ such subsets of layers (i.e., configurations). Let us denote a permutation of the numbers 1, 2, …, k with $\pi_1, \pi_2, \ldots, \pi_k$, such that $\pi_1, \pi_2, \ldots, \pi_\alpha$ are the levels of the multilayer network which contain the co-existing embeddings of the given motifs. Let us denote the set of all possible embeddings and the set of all nonoverlapping embeddings on the ith layer of the given multilayer network with $S_{F_1}^{(i)}$ and $S_{F_2}^{(i)}$. Thus, for each configuration, the final embedding set $S = \left(S_{F_2}^{(\pi_1)} \cap \cdots \cap S_{F_2}^{(\pi_\alpha)}\right) \setminus \left(S_{F_1}^{(\pi_{\alpha+1})} \cup \cdots \cup S_{F_1}^{(\pi_k)}\right)$. The first term computes the nonoverlapping embeddings that exist in the selected α layers and the second term removes those embeddings which appear in any of the remaining layers. After computing all configurations, we select the embedding set with the maximum size. We call this baseline method, the *Naive method* in the rest of this paper.

Implementation and System Details. We implement the algorithm in C++. We perform all the computational experiments on a Linux machine equipped with Intel core i7 processor 3.6 GHz CPU and 12GBs RAM.

3.1 Evaluation on synthetic datasets

We evaluate our method under a wide spectrum of parameters. We vary the network size, average degree and minimum frequency f. At each experiment, we vary one of these parameters and fix other two parameters. We conduct each experiment on 10 multi-layer networks. We measure the accuracy and running time for each method. We calculate the accuracy as the ratio of the number of embeddings discovered to the number of embeddings planted.

Effect of network size. First, we investigate the impact of network size. We use network sizes of 200, 400, 800 and 1600. We fix the average degree and minimum frequency f to 4 and 6 respectively. Figure 7(a) reports the results.

We first explore the effect on accuracy. We observe that our method achieves 100% accuracy for all network sizes and motif patterns. The accuracy of the Naive method gradually increases with growing networks sizes. One possible reason is that the networks become sparser with increasing network size when the degree of the network remains fixed. As the network gets sparser, the probability of generating other embeddings by randomly creating edges, which interferes with the co-existence of other embeddings, reduces. Moreover, the gap between our method and Naive method differs across four motif patterns. We conjecture that this is due to differences of motif topologies.

We observe that our method runs an order of magnitude faster than Naive method for all network sizes and motif patterns. The running time gradually increases when increasing network sizes. This is expected as the first step for both methods, finding all possible embeddings largely depends on the network sizes. In addition, both methods have practical running time, which implies that our method has great potential to scale to large networks.

Effect of network average degree. Next, we explore the impact of network density. We set the network size $|V|$ and minimum frequency f to 400 and 6 respectively. We vary the average degree from 2 to 10 at increments of 2. Figure 7(b) presents the results.

Consistent with the previous experiment, our method achieves 100% accuracy and runs at least an order of magnitude faster for all motif patterns and network average degrees as compared to the baseline algorithm. The accuracy of the Naive method, on the other hand, gradually goes down with growing network density. This is also consistent with the conjecture in the first experiment that sparser network tends to have smaller chance to miss the planted embeddings as more edges increases the chances of randomly generating new motif instances. Also, as the network density increases, the gap between the running time of our method and the baseline

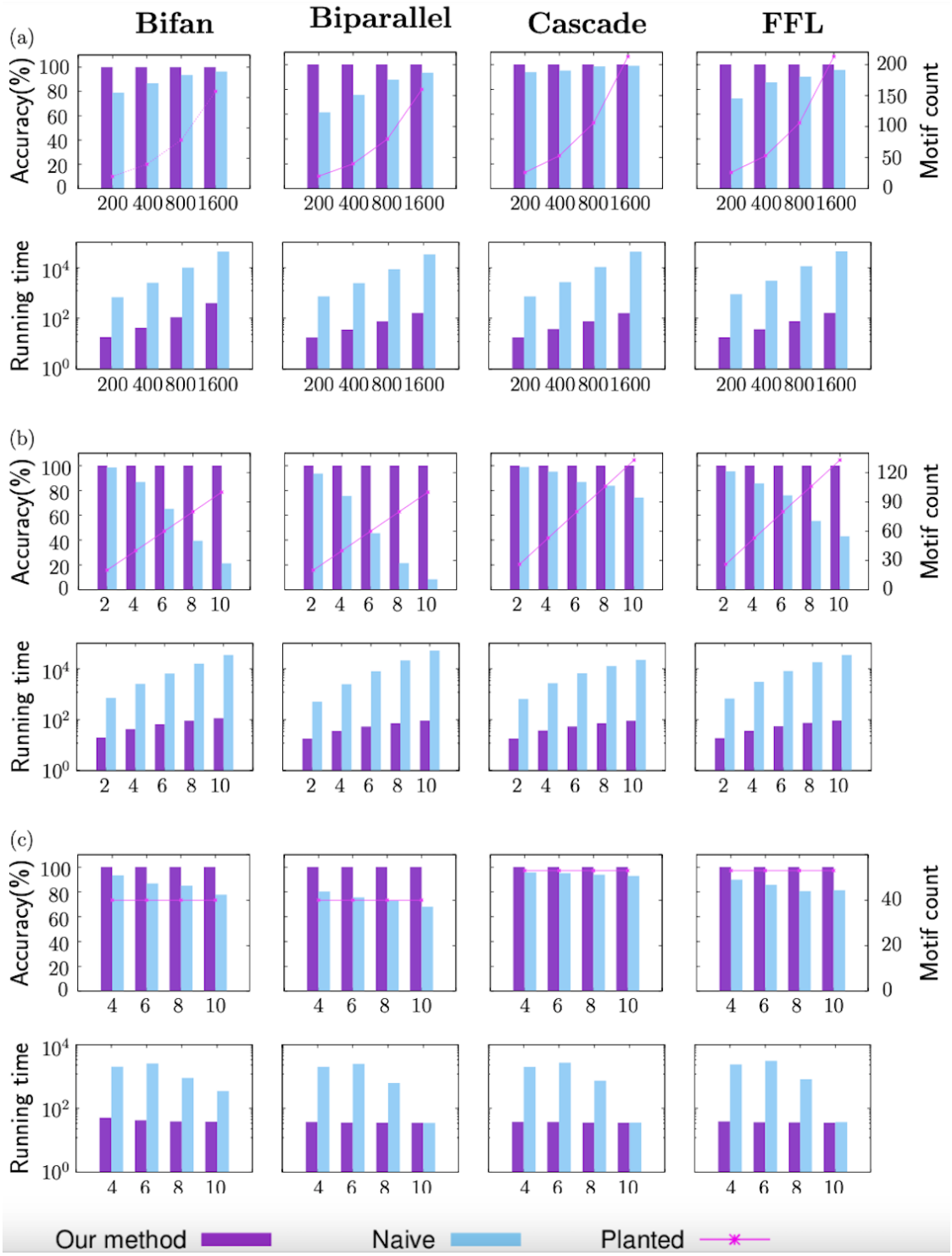


Figure 7: Accuracy and running time of methods on multilayer networks with various parameters for four motif patterns. The x-axis shows (a) network sizes; (b) network degrees; (c) minimum frequency f. The y-axis on the left of the first figure in (a) - (c) represents the accuracy of methods; the y-axis on the right represents the number of planted embeddings. The running times are reported in milliseconds and represented in log-scale.

algorithm grows.

Effect of minimum frequency f. Finally, we evaluate the impact of parameter f. Recall that small values of f indicate rare motif placements and large values of f indicate motif placements which are common across many network layers. We vary the frequency values from 4 to 10 at increments of 2. We set the network size to 400 and network average degree to 4. Figure 7(c) shows the results.

Similar to the previous experiments, our methods outperforms the Naive method in terms of both accuracy and running time across all frequency values and motif patterns. The accuracy of the Naive method gradually decreases with the increasing frequency values. This is because larger f values forces the existence of motifs in more network layers, making them harder to find. The running time of our method has no obvious change when increasing the frequency. As f grows, the running time of the Naive method first increases slightly, then decreases. This is because the number of subsets of f network layers out of k network layers has a binomial distribution. In the extreme case, when $f = k$ (i.e., $f = 10$ in our experiment), there is only one configuration as we require all network layers to contain the motif embeddings for $f = k$. Even when $f = k$, the running time of our method is the same as or better than the Naive method.

In summary, our method is efficient and has very high accuracy for all the parameter combinations we tested. While the network size and density influences the running time of our method, the distribution of motif embeddings across different network layers has no practical impact.

3.2 Evaluation on real datasets

In this section, we measure the performance of our method on three multi-layer transcriptional regulatory networks, namely human, yeast and E.coli.

Evaluation of motif count and running time. Similar to the synthetic experiments, we use the four popular functional motif patterns in Figure 6 and compare our method with the Naive method. As the true motif counts of these real networks are unknown, unlike synthetic dataset, we measure their performance in terms of motif count and running time. As the yeast and E.coli networks rarely contain cascade patterns, we do not report their result. We vary the value of f from 1 to total number of layers for each network and motif pattern. Figure 8 reports the result.

Figures 8(a) to 8(c) demonstrate that our method finds substantially more motifs than the Naive method. The gap between the two methods grow in favor of our method with increasing value of f. In addition, we observe that the motifs found by our method do not have obviously change with increasing frequency value. This implies that these motif patterns are conserved well across different conditions. One exception is for the biparallel motif in E.coli network. Notice that different motif count means totally different embedding set. Thus, we obtain four distinct embedding set when increasing f from 1 to 5. We investigate these embedding set later to see which conditions greatly affect these embeddings.

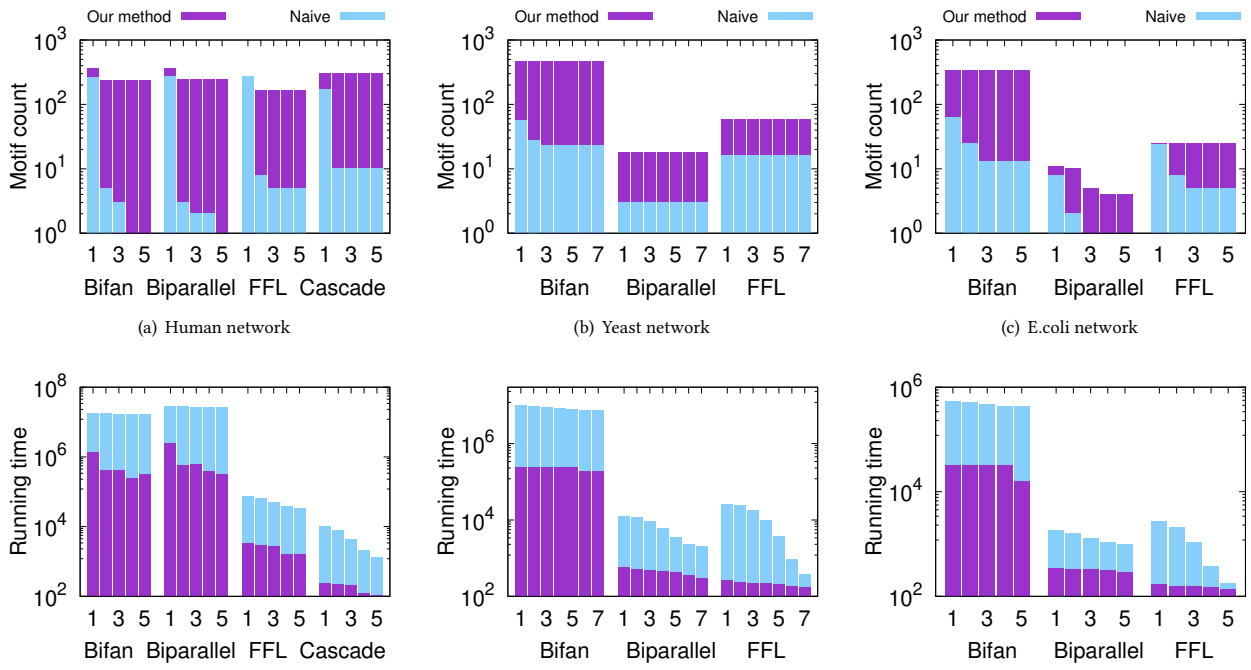
Figures 8(d) to 8(f) present the running times of the two methods. Consistent with our results on the synthetic dataset, our method runs at least an order of magnitude faster than the Naive method. The running times of both methods gradually go down with growing value of f, which implies that the large frequency value has the potential to filter more impossible embeddings comparing to the synthetic network. Moreover, we observe that our method still has the potential to scale to larger and denser networks as the largest running time is less than one hour.

Statistical significance of the result. We calculate the statistical significance of the results using *Z score*. In order to consider a suitable null-model, it is necessary to take into account the dependencies between layers. For such a reason, we randomize the edges as follows. Consider the aggregate graph $\mathcal{A} = (V, \mathcal{E}, \Omega)$. We construct a new aggregate graph $\mathcal{A}' = (V, \mathcal{E}', \Omega')$ guided by $\mathcal{A}$. The new aggregate graph has the same set of nodes as $\mathcal{A}$, but it initially has no edges. We then iterate over all edges of $\mathcal{A}$. For each edge (u, v) in $\mathcal{A}$, we randomly generate an edge (u', v'). Thus, we obtain the mappings of all original edges. We do not allow two different edges in $\mathcal{A}$ map to the same newly generated edge in $\mathcal{A}'$, and vice verse. For each edge $(u, v) \in \mathcal{E}$, we set $\Omega'(u', v') = \Omega(u, v)$. Thus, each layer in the newly generated multilayer network has the same number of interactions as well as the dependencies with the other layers. Assume that for each motif, the number of appearances in the real network is N_{real}; and in the randomized networks, the mean and standard deviation of the number of motifs are μ and σ respectively. We calculate the z-score as $Z = \frac{N_{real} - \mu}{\sigma}$. We think a motif is over-represented/under-represented if its Z score is $Z \geq 2$ / $Z \leq -2$. Figure 9 presents the results.

We observe that both bifan and feed forward loop are over-represented across three networks for almost all different minimum frequency values. Thus these two motif patterns are robust in various experimental conditions. Cascade pattern follows similar rules. Biparallel however has significant different behaviour across three networks. In human network, it is only over-represented when appearing all the layers. It is however over-represented across all layers in yeast network. As for the E.coli networks, it is over-represented only when appearing on smaller number of layers. Its significance however decreases with the increasing value of minimum frequency, which implies the stress conditions play a significant role on the generation of biparallel motifs. An interesting phenomenon is that in the human network, when the minimum frequency is equal to 1, all patterns are highly under-represented which suggests that all conditions have significant difference from each other.

4 CONCLUSION

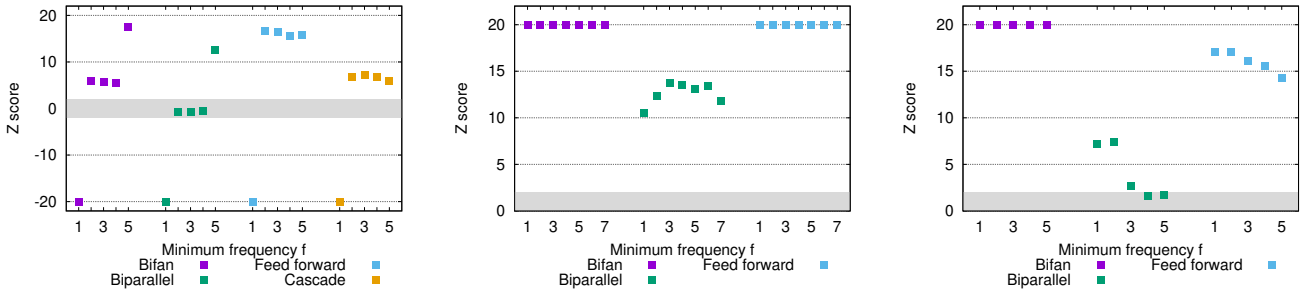
In this paper, we introduced the problem of co-existing motifs in multilayer networks. These motifs describe the dual conservation of the functions of cells within a network layer (i.e., cell condition) as well as across different layers of networks. We proposed a new algorithm to solve the co-existing motif identification problem efficiently and accurately. Our experiments on both synthetic and real datasets demonstrated that our method identifies all co-existing motifs at near 100 % accuracy for all networks we tested on, while competing method's accuracy varies greatly between 10 to 95 %. Furthermore, our method runs at least an order of magnitude faster



(a) Human network (b) Yeast network (c) E.coli network

(d) Human network (e) Yeast network (f) E.coli network

Figure 8: Number of motifs identified and the running time using our method and the Naive method on three gene regulatory networks; Human (a), (d); Yeast (b), (e); E.coli (c), (f). x-axis represents the value of minimum frequency f. The running times are reported in milliseconds and represented in log-scale.



(a) Human network (b) Yeast network (c) E.coli network

Figure 9: Z score of four motif patterns on three gene regulatory networks; (a) Human, (b) Yeast, (c) E.coli. We represent those large Z scores ($Z \geq 20$ or $Z \leq -20$) to be 20 or -20. The shaded area represents Z score values that are not significant. x-axis represents the value of minimum frequency f.

than state of the art motif identification methods for binary network models.

REFERENCES

[1] Federico Battiston, Vincenzo Nicosia, and Vito Latora. 2014. Structural measures for multiplex networks. *Physical Review E* 89, 3 (2014), 032804.

[2] Barry Bentley, Robyn Branicky, Christopher L Barnes, Yee Lian Chew, Eviatar Yemini, Edward T Bullmore, Petra E Vértes, and William R Schafer. 2016. The multilayer connectome of Caenorhabditis elegans. *PLoS computational biology* 12, 12 (2016), e1005283.

[3] Ginestra Bianconi. 2013. Statistical mechanics of multiplex networks: Entropy and overlap. *Physical Review E* 87, 6 (2013), 062806.

[4] Ginestra Bianconi and Sergey N Dorogovtsev. 2014. Multiple percolation transitions in a configuration model of a network of networks. *Physical Review E* 89, 6 (2014), 062814.

[5] Stefano Boccaletti, Ginestra Bianconi, Regino Criado, Charo I Del Genio, Jesús Gómez-Gardenes, Miguel Romance, Irene Sendina-Nadal, Zhen Wang, and Massimiliano Zanin. 2014. The structure and dynamics of multilayer networks. *Physics Reports* 544, 1 (2014), 1–122.

[6] M. Chen and R. Hofestaedt. 2006. *Prediction and Alignment of Metabolic Pathways.* Springer US, Boston, MA, 355–365. https://doi.org/10.1007/0-387-29455-4_34

[7] X. Chen, Natasha G. Deane, Keeli B. Lewis, Jiang Li, Jing Zhu, M. Kay Washington, and R. Daniel Beauchamp. 2016. Comparison of Nanostring nCounter® Data on FFPE Colon Cancer Samples and Affymetrix Microarray Data on Matched Frozen Tissues. *PLoS ONE* 11 (2016).

[8] Kevin Chow, Aisharjya Sarkar, Rasha Elhesha, Pietro Cinaglia, Ahmet Ay, and Tamer Kahveci. 2021. <italic>ANCA</italic>: Alignment-Based Network Construction Algorithm. *IEEE/ACM Transactions on Computational Biology and Bioinformatics* 18, 2 (2021), 512–524. https://doi.org/10.1109/TCBB.2019.2923620

[9] Diane J. Cook and Lawrence B. Holder. 1994. Substructure discovery using minimum description length and background knowledge. *J. Artif. Int. Res.* 1, 1 (Feb. 1994), 231–255.

[10] Stephen A Cook. 1971. The complexity of theorem-proving procedures. In *Proceedings of the third annual ACM symposium on Theory of computing*. ACM, 151–158.

[11] Rasha Elhesha and Tamer Kahveci. 2016. Identification of large disjoint motifs in biological networks. *BMC Bioinformatics* 17, 408 (2016).

[12] Sergio Gomez, Albert Diaz-Guilera, Jesus Gomez-Gardenes, Conrad J Perez-Vicente, Yamir Moreno, and Alex Arenas. 2013. Diffusion dynamics on multiplex networks. *Physical review letters* 110, 2 (2013), 028701.

[13] Jesús Gómez-Gardenes, Irene Reinares, Alex Arenas, and Luis Mario Floría. 2012. Evolution of cooperation in multiplex networks. *Scientific reports* 2 (2012), 620.

[14] Michelle L Green and Peter D Karp. 2004. A Bayesian method for identifying missing enzymes in predicted metabolic pathway databases. *BMC bioinformatics* 5, 1 (2004), 1.

[15] Haiyan Hu, Xifeng Yan, Yu Huang, Jiawei Han, and Xianghong Jasmine Zhou. 2005. Mining coherent dense subgraphs across massive biological networks for discovery. *Bioinformatics* 21 (January 2005), 213–221. Issue 1.

[16] Yanhui Hu, Ian Flockhart, Arunachalam Vinayagam, Clemens Bergwitz, Bonnie Berger, Norbert Perrimon, and Stephanie E Mohr. 2011. An integrative approach to ortholog prediction for disease-focused and other functional studies. *BMC bioinformatics* 12, 1 (2011), 1.

[17] Jun Huan, Wei Wang, and Jan Prins. 2003. Efficient Mining of Frequent Subgraphs in the Presence of Isomorphism. In *ICDM*. 549–552.

[18] Szymon Jozefczuk, Sebastian Klie, Gareth Catchpole, Jedrzej Szymanski, Alvaro Cuadros-Inostroza, Dirk Steinhauser, Joachim Selbig, and Lothar Willmitzer. 2010. Metabolomic and transcriptomic stress response of Escherichia coli. *Molecular systems biology* 6, 1 (2010), 364.

[19] Brian P. Kelley, Roded Sharan, Richard M. Karp, Taylor Sittler, David E. Root, Brent R. Stockwell, and Trey Ideker. 2003. Conserved pathways within bacteria and yeast as revealed by global protein network alignment. *Proceedings of the National Academy of Sciences of the United States of America* 100 (2003), 11394 – 11399.

[20] Jung Yeol Kim and K-I Goh. 2013. Coevolution and correlated multiplexity in multiplex networks. *Physical review letters* 111, 5 (2013), 058702.

[21] Mikko Kivelä, Alex Arenas, Marc Barthelemy, James P Gleeson, Yamir Moreno, and Mason A Porter. 2014. Multilayer networks. *Journal of complex networks* 2, 3 (2014), 203–271.

[22] Mehmet Koyutürk, Ananth Grama, and Wojciech Szpankowski. 2004. An efficient algorithm for detecting frequent subgraphs in biological networks. *Bioinformatics* 20 (january 2004), 200–207. Issue 1.

[23] Michihiro Kuramochi and George Karypis. 2004. GREW-A Scalable Frequent Subgraph Discovery Algorithm. In *ICDM*. 439–442.

[24] Tong I. Lee, Nicola J. Rinaldi, François Robert, Duncan T. Odom, Ziv Bar-Joseph, Georg K. Gerber, Nancy M. Hannett, Christopher T. Harbison, Craig M. Thompson, Itamar Simon, Julia Zeitlinger, Ezra G. Jennings, Heather L. Murray, D. Benjamin Gordon, Bing Ren, John J. Wyrick, Jean-Bosco Tagne, Thomas L. Volkert, Ernest Fraenkel, David K. Gifford, and Richard A. Young. 2002. Transcriptional regulatory networks in Saccharomyces cerevisiae. *Science* 298, 5594 (2002), 799–804.

[25] Wenyuan Li, Chun-Chi Liu, Tong Zhang, Haifeng Li, Michael S Waterman, and Xianghong Jasmine Zhou. 2011. Integrative analysis of many weighted co-expression networks using tensor computation. *PLoS computational biology* 7, 6 (2011), e1001106.

[26] Virginia D Marks, Shannan J. Ho Sui, Daniel J. Erasmus, George van der Merwe, Jochen Brumm, Wyeth W. Wasserman, Jennifer Bryan, and Hennie J. J. van Vuuren. 2008. Dynamics of the yeast transcriptome during wine fermentation reveals a novel fermentation stress response. *FEMS yeast research* 8 1 (2008), 35–52.

[27] Giulia Menichetti, Daniel Remondini, Pietro Panzarasa, Raúl J Mondragón, and Ginestra Bianconi. 2014. Weighted multiplex networks. *PloS one* 9, 6 (2014), e97857.

[28] Tom Michoel and Bruno Nachtergaele. 2012. Alignment and integration of complex networks by hypergraph-based spectral clustering. *Physical Review E* 86, 5 (2012), 056111.

[29] T. Milenković, W. Ng, W. Hayes, and N. Pržulj. 2010. Optimal network alignment with graphlet degree vectors. *Cancer Informatics* 9 (2010), 121.

[30] R Milo, N Kashtan, S Itzkovitz, MEJ Newman, and U Alon. 2003. On the uniform generation of random graphs with prescribed degree sequences. *arXiv preprint cond-mat/0312028* (2003).

[31] Ron Milo, Shai Shen-Orr, Shalev Itzkovitz, Nadav Kashtan, Dmitri Chklovskii, and Uri Alon. 2002. Network motifs: simple building blocks of complex networks. *Science* 298, 5594 (2002), 824–827.

[32] Shane Neph, Andrew B Stergachis, Alex Reynolds, Richard Sandstrom, Elhanan Borenstein, and John A Stamatoyannopoulos. 2012. Circuitry and dynamics of human transcription factor regulatory network s. *Cell* 150, 6 (2012), 1274–1286.

[33] Vincenzo Nicosia, Ginestra Bianconi, Vito Latora, and Marc Barthelemy. 2013. Growing multiplex networks. *Physical review letters* 111, 5 (2013), 058701.

[34] Vincenzo Nicosia and Vito Latora. 2015. Measuring and modeling correlations in multiplex networks. *Physical Review E* 92, 3 (2015), 032805.

[35] Vincenzo Nicosia, Miguel Valencia, Mario Chavez, Albert Díaz-Guilera, and Vito Latora. 2013. Remote synchronization reveals network symmetries and functional modules. *Physical review letters* 110, 17 (2013), 174102.

[36] Siegfried Nijssen and Joost N. Kok. 2004. A quickstart in frequent structure mining can make a difference. In *KDD* (Seattle, WA, USA). 647–652.

[37] Mostafa Salehi, Rajesh Sharma, Moreno Marzolla, Matteo Magnani, Payam Siyari, and Danilo Montesi. 2015. Spreading processes in multilayer networks. *IEEE Transactions on Network Science and Engineering* 2, 2 (2015), 65–83.

[38] Alberto Santos-Zavaleta, Heladia Salgado, Socorro Gama-Castro, Mishael Sanchez-Parez, Laura Gómez-Romero, Daniela Ledezma-Tejeida, Jair Santiago Garcia-Sotelo, Kevin Alquicira-Hernandez, Luis Jose Muniz-Rascado, Pablo Pena-Loredo, Cecilia Ishida-Gutierrez, David A Velazquez-Ramirez, Víctor Del Moral-Chavez, Cesar Bonavides-Martinez, Carlos-Francisco Mendez-Cruz, James Galagan, and Julio Collado-Vides. 2018. RegulonDB v 10.5: tackling challenges to unify classic and high throughput knowledge of gene regulation in E. coli K-12. *Nucleic Acids Research* 47, D1 (11 2018), D212–D220.

[39] A. Sarkar, Y. Ren, R. Elhesha, and T. Kahveci. 2019. A New Algorithm for Counting Independent Motifs in Probabilistic Networks. *IEEE/ACM Transactions on Computational Biology and Bioinformatics* 16, 4 (2019), 1049–1062.

[40] Jacob Scott, Trey Ideker, Richard M Karp, and Roded Sharan. 2006. Efficient algorithms for detecting signaling pathways in protein interaction networks. *Journal of Computational Biology* 13, 2 (2006), 133–144.

[41] Shai S Shen-Orr, Ron Milo, Shmoolik Mangan, and Uri Alon. 2002. Network motifs in the transcriptional regulation network of Escherichia coli. *Nature Genetics* 31, 1 (2002), 64–68.

[42] Luis Solá, Miguel Romance, Regino Criado, Julio Flores, Alejandro García del Amo, and Stefano Boccaletti. 2013. Eigenvector centrality of nodes in multiplex networks. *Chaos: An Interdisciplinary Journal of Nonlinear Science* 23, 3 (2013), 033131.

[43] Miguel C Teixeira, Pedro T Monteiro, Margarida Palma, Catarina Costa, Cláudia P Godinho, Pedro Pais, Mafalda Cavalheiro, Miguel Antunes, Alexandre Lemos, Tiago Pedreira, and Isabel Sá-Correia. 2017. YEASTRACT: an upgraded database for the analysis of transcription regulatory networks in Saccharomyces cerevisiae. *Nucleic Acids Research* 46, D1 (09 2017), D348–D353.

[44] Pei Wang, Jinhu Lü, and Xinghuo Yu. 2014. Identification of important nodes in directed biological networks: A network motif approach. *PloS One* 9, 8 (2014), e106132.

[45] Sebastian Wernicke. 2005. A faster algorithm for detecting network motifs. In *LNBI 3692*. Springer-Verlag, 165–177.

[46] Tzu-Hsien Yang, Chung-Ching Wang, Yu-Chao Wang, and Wei-Sheng Wu. 2014. YTRP: a repository for yeast transcriptional regulatory pathways. *Database* 2014 (03 2014).

[47] Li Zhenping, Shihua Zhang, Yong Wang, Xiang-Sun S. Zhang, and Luonan Chen. 2007. Alignment of molecular networks by integer quadratic programming. *Bioinformatics* 23 (2007), 1631–1639.