Trend Motif: A Graph Mining Approach for Analysis of Dynamic Complex Networks

Ruoming Jin, Scott McCallen
Department of Computer Science,Kent State University, Kent, OH, 44241
{jin,smccalle}@cs.kent.edu

Eivind Almaas
Microbial Systems Group, Biosciences & Biotechnology Division,
Lawrence Livermore National Laboratory, Livermore, CA 94551-0808
almaas@llnl.gov

Abstract

Complex networks have been used successfully in scientific disciplines ranging from sociology to microbiology to describe systems of interacting units. Until recently, studies of complex networks have mainly focused on their network topology. However, in many real world applications, the edges and vertices have associated attributes that are frequently represented as vertex or edge weights. Furthermore, these weights are often not static, instead changing with time and forming a time series. Hence, to fully understand the dynamics of the complex network, we have to consider both network topology and related time series data.

In this work, we propose a motif mining approach to identify trend motifs for such purposes. Simply stated, a trend motif describes a recurring subgraph where each of its vertices or edges displays similar dynamics over a user-defined period. Given this, each trend motif occurrence can help reveal significant events in a complex system; frequent trend motifs may aid in uncovering dynamic rules of change for the system, and the distribution of trend motifs may characterize the global dynamics of the system. Here, we have developed efficient mining algorithms to extract trend motifs. Our experimental validation using three disparate empirical datasets, ranging from the stock market, world trade, to a protein interaction network, has demonstrated the efficiency and effectiveness of our approach.

1. Introduction

The study of complex networks has emerged into an active interdisciplinary research field. Many complex systems, spanning from the structures of the Internet, human social networks, to gene-regulatory "circuitry" in single cells, have been constructed in the last several years. Surprisingly, these very different networks have several important features in common, such as a "scale-free" degree distribution and being "small-world." Various mathematical models have been proposed that give rise to such properties [4], and different clustering or decomposition methods have been developed to identify sets of small building blocks that highlight network design principles. In particular *network motifs*, which can be loosely described as over-represented subgraphs, have been demonstrated to yield significant insight into the composition and function of networks in biochemistry, neurobiology, ecology, and engineering [19, 20].

Recently, the dynamic processes taking place in complex networks have attracted much attention. This is because a complex network in the real world usually corresponds to an evolving system in a state of constant change. Often, these systems have been described by various epidemic modeling approaches, e.g. to simulate the diffusion of innovation, or to prevent and suppress the spread of computer viruses and sexually transmitted diseases, among others [7].

1.1 Motivation

The majority of recent studies have focused on characterizing the topology, or the change in topology, of complex networks [17, 5, 24]. However, in many real world applications, weights are often associated with the vertices or edges of the network. These weights are typically changing with time, thus forming a time series for each vertex and edge. Thus, knowledge of the network topology, paired with the time series data, provides a comprehensive global picture of a dynamically changing system. Generally speaking, if each vertex of the network has a weight, we refer to it as a vertex-weighted network or graph, and if each edge of the network has a weight, we refer to it as a edge-weighted network or graph. Note that, a network can be both vertex-weighted and edge-weighted.

In the following, we will consider several systems that can naturally be represented as weighted networks, and where the system dynamics are captured in time series of weights.

Financial Market: In the financial market, companies interact with each other and form various relationships, typically including competitor, producer-consumer, ownership, etc. A complex network can be built to represent the interactions of all the companies in the financial market, where each company corresponds to a vertex, and the relationship between two companies corresponds to an edge. Each vertex (company) can be weighted by the corresponding time series of stock value. Since the price change of each stock is often correlated with or determined by the price changes of companies with which it has close relations, the network representation provides a framework to simultaneously analyze the dynamics of an entire financial market.

Collaboration Network: Collaborations between scientists, as reflected in co-authorship on publications, is one of the widely studied subjects in the field of social networks and complex network mining. Here, each vertex represents a scientist, and two vertices (scientists) are connected with an edge if they co-authored a paper. The strength of the collaboration can be estimated by, e.g. the number of papers the scientists have co-authored in a given time frame. In the network representation, the measure of collaboration intensity can be represented as an edge-weight time series.

Protein Interaction Network: In the recent era of systems biology, new experimental approaches have been developed with the ability to rapidly measure thousands of molecular interactions. Among the most heralded are the so-called high-throughput techniques to characterize all pairs of proteins with the ability to physically interact. It has become customary to represent the resulting datasets as networks, where each vertex corresponds to a protein and two vertices are connected by an edge if the corresponding proteins can bind. In addition, the high-throughput microarray technology allow biologists to measure the distribution of gene products at different conditions and different time points. Thus, associating a time series from the microarray experiment for each protein provides a more comprehensive picture of the dynamically changing system inside a cell.

While we have focused on the possibility that the network weights will change with time in response to a system's dynamic processes, the topology of the underlying network may change as well. However, for many systems the typical time scale for weight dynamics is significantly shorter than that of the changes in the network topology. Consequently, it is reasonable to consider the network as a static entity. We note that, while a network with time-varying weights contains significantly more information about the system, few methods have been devised that leverage this information. Specifically, scientists would like to know what are the basic rules that govern the evolution and changes of the complex system, and how two different dynamic systems can be compared.

1.2 Our approach

Our approach to analyze the dynamic complex network starts from local dynamics. It is based on the observation that the weight change of a vertex in a complex network is rarely an isolated event. They are often strongly correlated with, or possibly determined by, the changes occuring in its network neighbors. Similar observation can be made for the edges as well. For instance, in the stock market, the increase of the Intel stock price is likely correlated with the increase (or decrease) of AMD's stock, and both correlate with the stock price of PC producers such as HP and Dell. Similarly, in the protein interaction network, a biological process is very likely to result in the co-changes of several related proteins [28]. In other words, synchronized changes of weights over closely related vertices or edges can serve as a good indication of (local) dynamics or the evolution of a system.

The central theme of this paper is the introduction and discovery of *trend motifs*, which target putative patterns of changes for a group of closely related entities. Given a weighted (undirected) complex network, a group of such entities corresponds to a set of connected vertices. A possible pattern of change (*a trend motif occurrence*) is a set of connected vertices associated with a time span where the time series of each vertex displays a consistent *trend*. Here, we focus on two types of trends: the first corresponds to a steady increase in the time series, and the second corresponds to a steady decrease (see Section 2 for the formal definition). Consequently, a putative pattern is likely to correspond to a major event, or a sequence of events, occurring in the system. Therefore, extracting such patterns can help scientists identify such events, which often are hidden in large amounts of data.

Further, we define a *frequent trend motif* as a putative pattern which are over-represented in the complex network. Frequent trend motifs can help reveal the underlying mechanism governing the dynamics. For instance, a line subgraph with each vertex showing increase may correspond to a cascade in the system, and a clique subgraph with some vertices showing increase with others showing a decrease may indicate these changes are strongly correlated. Finally, we note that the distribution of trend motifs can be used to categorize the dynamic networks, as we can expect that different types of networks will tend to have different types and distributions of such motifs [20].

Our contribution in the paper is as follows.

1. We formally introduce the concept of trend motif. To the best of our knowledge, this is the first work which applies a motif/subgraph mining approach to study the dynamics in a complex network.
2. We develop a flexible framework and several novel algorithms to efficiently discover these trend motifs.
3. We demonstrate the effectiveness and efficiency of our

approach through a detailed experimental evaluation using three empirical datasets on (i) a financial market [22, 23], (ii) global trade and GDP [12], and (iii) a protein interaction network [6] with associated microarray mRNA expression data [28].

2. Problem Definition

As previously discussed, we can intuitively understand a trend motif to be a recurring subgraph which, over time, displays a consistent pattern of increasing or decreasing weights of the vertices or edges. As two examples of possible trend motifs, consider an interval of the time series for which a K_3-clique has two vertices with increasing weights and one vertex with decreasing weight, or a K_5-clique has three vertices with decreasing weights and two vertices with increasing weights.

In the following, we will formally introduce the notation of trend motif. We note that the discussion will focus only on the vertex-weighted graphs for the sake of simplicity, and the graph notation will be formally used to describe the complex networks.

2.1. Trends and Trend Intervals

Given a graph $G = (V, E)$ of N vertices $V = \{v_1, v_2, \cdots, v_N\}$ and a discrete time span $[1, T]$, the weight of vertex v_i is denoted as $x_i(t)$, for $t \in [1, T]$. Intuitively, we consider a trend as a subsequence of a time series that shows a consistent increase or decrease. Formally, we define an **increasing trend** as a subsequence $[x_i(t_1), x_i(t_2), \cdots, x_i(t_k)]$, and $t_j < t_{j+1}$, of the time series $x_i(t)$ with respect to two parameters δ and σ, and it satisfies the following two conditions:

1. **Weight constraints:** for any time t_j, $x_i(t_{j+1}) - x_i(t_j) \geq \delta$, $\delta > 0$;
2. **Step constraints:** for two time points in the subsequence, $t_{j+1} - t_j \leq \sigma, \sigma > 0$.

Essentially, the movement threshold, δ, means the series has to continue to make at least δ change over the entire span of the trend, and the time step constraint, σ, means that the change δ can not be on opposite ends of the time series, but has to occur within a shorter amount of time. These conditions impose that the time series must contain a consistent increase (δ) within a specified amount of time (σ) in order to be identified as containing an increasing trend. Similarly, we define a **decreasing trend** as a subsequence of $x_i(t)$ so that, for any time t_j, $x_i(t_{j+1}) - x_i(t_j) \leq -\delta$ and $t_{j+1} - t_j \leq \sigma, \delta > 0, \sigma > 0$. If a subsequence satisfies one of these two definitions, either increasing or decreasing, it will simply be called a **trend**.

As a running example, consider the time series $x_i(t) = [1, 2, 3, 7, 5, 4, 12, 14, 13, 13, 15]$ with a threshold parameter of $\delta = 1$ and a maximum time jump parameter of $\sigma = 2$. Among the possible trends in this time series, we can easily identify the two sets $[1, 2, 3, 7]$ and $[4, 12, 13]$. Note that $[1, 2, 3]$ is also a trend. We define a **maximal trend** as a trend that is not a subset of any other trend in the time series, i.e. $[1, 2, 3, 7]$ is a maximal trend while $[1, 2, 3]$ is not. Formally, a trend S is a maximal trend if $\nexists S' | S \subset S'$.

To facilitate our discussion, we define an interval $[t_s, t_e]$ to be an **increasing trend interval** if it contains a trend $[x_i(t_s), \cdots, x_i(t_e)]$, where t_s and t_e are the beginning and ending points of the trend, respectively. We use the notation $[t_s, t_e]+$ to represent an increasing trend interval from start time t_s to end time t_e. Similarly, we can define the **decreasing trend interval** and denote it as $[t_s, t_e]-$. From the example, we can see the increasing trend interval $[1, 8]+$ which contains the trend $[1, 3, 5, 12, 14]$. In addition, we have $[1, 11]+$ as an increasing trend, since $[1, 2, 3, 5, 12, 13, 15]$ satisfies the conditions ($\delta = 1, \sigma = 2$). We note that for the first interval $[1, 8]+$ is a sub-interval of the latter one $I[1, 11]+$. Thus, we define the maximal interval of trend as the longest time span in $\{x_i(t)\}$ such that the values are consistently increasing or decreasing according the definition of a trend. Formally, we refer to $[t_s, t_e]$ as a **maximal interval of increasing trend** if it is an increasing trend interval $[t_s, t_e]+$ and there are no $[t'_s, t'_e]+$, such that $[t_s, t_e] \subset [t'_s, t'_e]$. The **maximal interval of decreasing trend** can be defined similarly.

2.2. Trend Motif

Given the previous definitions, we can identify the trends which indicate the increasing and/or decreasing intervals for each of the vertices individually over the entire time series. A particularly interesting pattern, however, is observed when multiple trends occur simultaneously, and especially when they occur in nodes that are closely related through the network topology. To properly describe this phenomenon, we will formally introduce the concept of *trend motif occurrence*. Given the graph $G = (V, E)$ and a subset of vertex $V_s \subset V$, let $G(V_s)$ be the *induced subgraph* of V_s [11]. Mathematically, the induced subgraph of V_s, $G(V_s)$, contains all the edges in E that have both ends in V_s.

Definition 1 Trend Motif Occurrence: *Given a graph G, a trend motif occurrence of G is defined as the triple $(V_s, [t_s, t_e], f)$ with ($t_s < t_e$), where $G(V_s)$ is a connected subgraph, f is a function $f : V_s \to \{+, -\}$, and $[t_s, t_e] = [t_s^1, t_e^1] \cap [t_s^2, t_e^2] \cap \cdots \cap [t_s^n, t_e^n]$, where $[t_s^i, t_e^i]$ is a maximal interval of trend for vertex $v_i \in V_s$, and n is the number of vertices in V_s.*

Note that, if $f(v_i) = +$, the corresponding interval is increasing, otherwise $f(v_i) = -$. Basically, the function f labels each node of $G(V_s)$. We denote the labeled graph as $G^f(V_s)$. Additionally, we note that the interval $[t_s, t_e]$ is the intersection of all maximal intervals of trend, and the intersection of the maximal intervals on $[t_s, t_e]$ has to be nonempty. However, this intersection need not be a maximal trend interval on any of the vertices in V_s.

Based on the above definition, a very large number of trend motif occurrences may exist in a complex network for any time span. To reduce the number of motif occurrences, we introduce two parameters l and w, where l is the minimum interval length for a trend interval of each vertex in the motif occurrence and w is the minimal length for the intersection of the motif occurrence. We denote such a trend motif occurrence given l and w as $(V_s, [t_s, t_e], f)(l, w)$.

Finally, we introduce the concept of a *frequent trend motif*. Given two trend motif occurrences, $(V_1, [t_s^1, t_e^1], f_1)$, and $(V_2, [t_s^2, t_e^2], f_2)$, $V_1 \neq V_2$, we refer to them as *equivalent* if their corresponding labeled induced subgraphs are isomorphic $G^{f_1}(V_1) = G^{f_2}(V_2)$ [11]. In other words, there exists a one-to-one mapping between V_s and V_s', $g : V_s \rightarrow V_s'$, such that for any $v_i, v_j \in V_s$, $(v_i, v_j) \in E(G(V_s)) \Leftrightarrow (g(v_i), g(v_j)) \in E(G(V_s'))$, and $f_1(v_i) = f_2(g(v_i))$. Here $E(G(V_s))$ and $E(G(V_s'))$ are the the edge sets of the induced graph of $G(V_s)$ and $G(V_s')$, respectively.

Definition 2 Frequent Trend Motif: *Given a support θ, and two parameters l and w, if there are more than or equal to θ distinct subset of vertices, $V_1, \cdots, V_t$, $t \geq \theta$, such that each set has at least a trend motif occurrence $(V_i, [t_s^i, t_e^i], f_i)(l, w)$ being equivalent, then we refer to $G_s^f(l, w, \theta)$ as a frequent trend motif, where G_s^f is a labeled subgraph that is isomorphic to $G^{f_i}(V_i), 1 \leq i \leq t$.*

Consequently, we can identify the following two related mining tasks.

1. **Extracting Trend Motif Occurrences:** Given two parameters l and w, we would like to find all the trend motif occurrences $(V_s, [t_s, t_e], f)(l, w)$ in a graph G.
2. **Extracting Frequent Trend Motifs:** Given the support level θ and the parameters l and w, we would like to find all the frequent trend motifs $G_s^f(l, w, \theta)$.

Clearly, these mining tasks are different from traditional subgraph mining tasks [14, 15, 16]. In the subgraph mining, the label of each vertex is known, and the major task is to enumerate all the possible candidate subgraphs, counting their number of occurrences. Here, each motif occurrence is dynamically determined by the time series data. In addition, each induced subgraph may correspond to different types of trend motif occurrences, as each vertex may display different trends at different time points. If we label each vertex with either $+$ (corresponding to increasing trend intervals) or $-$ (corresponding to decreasing trend intervals), a vertex can have both labels. These considerations show that mining trend motifs is a challenging task.

3. Algorithms

In this section, we will introduce efficient algorithms for the two mining tasks, extracting all the trend motif occurrences and extracting all the frequent trend motifs. The basic idea of our approach for these two mining tasks is as follows. We will first extract all the maximal intervals of trends for each vertex, and organize them into two categories, corresponding to the increasing trend and the decreasing trend. Then, we will use the depth-first approach to traverse the underlying graph to find any induced subgraph that are associated with trend intervals which satisfy the two length constraints l and w. Finally, we will use a level-wise approach to find all the frequent motifs using the discovered motif occurrences.

We will first present an algorithm for extracting maximal intervals of trends in Subsection 3.1, which will be the basis for these two mining tasks. Then, we will introduce the algorithm in Subsection 3.2 for the first mining task. In Subsection 3.3, we will discuss the algorithm which will use the result from the first task to extract frequent trend motifs.

3.1 Extracting Maximal Trend Intervals

Consider we have a time series $X(t), t \in [1, T]$ and two parameters δ and σ, we would like to extract all the maximal trend intervals from $X(t)$. A simple attempt will be to extract all the maximal trends first and then generate intervals defined by the starting time point and the end time point of these maximal trends. However, this approach can be rather computationally expensive. First, we note that the maximal trend intervals are not necessarily the *the maximal intervals of trends*. Thus, a much larger number of maximal trends which will not correspond to the maximal intervals of trends can be generated. Therefore, our approach tries to directly generate these maximal intervals of trends.

Algorithm 1 $ExtractTrendIntervals(\delta, \sigma, X)$

```
$Q \leftarrow \emptyset$ { sorted list holds the last $\sigma$ elements seen}
for $t = 1$ to $|X|$ do
    $inc(t) \leftarrow min\{inc(q)|X(q) + \delta \leq X(t), X(q) \in Q\}$ {inc(t) is the earliest time that [inc(t),t] is an interval of increasing trend}
    $dec(t) \leftarrow min\{inc(q)|X(q) \geq X(t) + \delta, X(q) \in Q\}$ {dec(t) is the earliest time that [dec(t),t] is an interval of decreasing trend}
    $Q \leftarrow Q \cup \{X(t)\}$ {add to the queue}
    if $|Q| > \sigma$ then
        $Q \leftarrow Q \setminus X(t - \sigma)$ {remove the earliest}
        if $\forall\, X(q) \in Q, inc(q) > inc(t - \sigma)$ then
            $interval[+] \leftarrow interval[+] \cup \{[inc(t - \sigma), t - \sigma]\}$
        end if
        if $\forall\, X(q) \in Q, dec(q) > dec(t - \sigma)$ then
            $interval[-] \leftarrow interval[-] \cup \{[dec(t - \sigma), t - \sigma]\}$
        end if
    end if
end for
return interval;
```

Here, we introduce an algorithm with a linear time complexity to simultaneously extract all maximal intervals of both increasing and decreasing trends in one pass through a time series. The *ExtractTrendIntervals* algorithm is shown in Algorithm 1. The algorithm maintains a list Q that stores the last σ seen elements at any time point t, from the given time series X. We iteratively look at each of the n elements in X_i (The for loop at line 2). The key of this algorithm is for each time point t, we will derive two values, $inc(t)$ and $dec(t)$, which correspond to the intervals of increasing trend and decreasing trend, respectively. Essentially, $inc(t)$ is the earliest time point which can form an interval of increasing trend together with t. This is equivalent to say that $[inc(t), t]$ is the longest interval which contains an increasing trend starting from $inc(t)$ and end at the current time point t. This is achieved by appending $X(t)$ to all the elements in Q, which satisfy the *weight increasing constraint* (Subsection 2.1) between $X(t)$ and $X(q), q \in Q$. Among those satisfying the constraint, we will choose the one which has the earliest time point forming the interval of increasing trends (Line 3). The processing for $dec(t)$ is similar (Line 4).

Given this, for a time point t, we basically have the longest intervals of trends which ends with t. The next question will be under what condition, such longest intervals will become *maximal intervals* of trends. We begin testing if there is a maximal interval ending with t when the Q is full. In other words, we drop the element $X(t)$ when the $t + q$ time point arrives. This is because starting from $t + q$, no other time point will be able to directly connect to $X(t)$ to form a trend based on the *step constraint* (Subsection 2.1). The condition for ensuring the maximal intervals of trends is rather simple: we basically want to see if the element being removed has a trend interval that is not a subset of any other trend interval in Q. This can be simply achieved through condition in Line 8 for increasing trends and Line 11 for decreasing trends. This can easily ensure that no $X(q) \in Q$ or $(t - \sigma < q \leq t)$, such that $[inc(t-\sigma), t-\sigma] \subset [inc(q), q]$. Based on the above discussion, we can have the following lemma stating the correctness of our algorithm.

Lemma 1 *Given parameters, δ and σ, the algorithm ExtractTrendIntervals will extract all the maximal intervals of both increasing trends and decreasing trends from the input time series X.*

Finally, we note that the computational complexity of this algorithm is $|X|\sigma$. This is because for each time point t, we have to build $inc(t)$ and $dec(t)$. These two operations will require an upper bound of $O(|Q|) = O(\sigma)$ time complexity. Also, this is a one pass algorithm which requires only $O(\sigma)$ space complexity. Thus, it can be applied to streaming data.

3.2 Algorithm for Trend Motif Occurrence Discovery

One of the major difficulties in enumerating all the trend motif occurrences is the massive search space which spans both the topology dimension and the time dimension: any subset of connected vertices (topology dimension) combining with an interval (time dimension) can be treated as a candidate of trend motif occurrence. However, only a small portion of these candidates will become the true occurrences.

In order to efficiently discover these motif occurrences, we have to aggressively prune the search space. Here, we apply several techniques to reduce the search space. The first technique is based on the down-closure property: for any motif occurrence $(V_s, [t_s, t_e], f)$, any subset of connected vertices $V'_s \subseteq V_s$ will correspond to a motif occurrence whose interval contains $[t_s, t_e]$. This will enable us to apply a depth-first search strategy to enumerate the motif occurrences from a single vertex to larger patterns. Secondly, we will enumerate all the motif occurrences which correspond to the same subset of vertices V_s and share the same labeling function f together. We refer to these motif occurrences as the same *type* of motif occurrences. This essentially enables us to enumerate the same type of motif occurrences in an efficient way.

Further, to reduce the cost of trend interval discovery, we extract all the maximal intervals of both increasing trends and decreasing trends for each vertex in the graph G using *ExtractTrendIntervals*. Then, for each vertex v, we record all the maximal intervals of increasing trends and decreasing trends (whose lengths are no less than l) in $v.interval[+]$ and $v.interval[-]$, respectively. Thus, we discover all the intervals of trends for each vertex only once. In addition, if a vertex does not have any interval, we remove them from the original graph G. This can help to reduce the search space.

The key procedure in enumerating the trend motif occurrence is illustrated in the $Build$ method (Algorithm 2), which employs a depth-first search (DFS) strategy. All the occurrences are recorded in a tree structure. Each node of the tree corresponds to a vertex with certain trend, increasing (+) or decreasing (-). A path starting from the root to the given node v encodes one type of motif occurrence, and this node also records all the trend intervals of this type of motif occurrence in $v.interval$. The $Build()$ operation begins with a root node r that has no children, a set of neighbors N of the current motif occurrences and an excluded set E that records which vertices can no longer joined to the current occurrence. Both of the sets are initially empty ($Build(r, \emptyset, \emptyset)$). In addition, we assume the root node r has all the vertex in G as its neighbors: $Neighbor(v) = V(G)$, and $r.interval$ records only one interval $[1, \infty]$, suggesting

Algorithm 2 $Build(Node\ v, Set\ N, Set\ E)$

```
1: $N \leftarrow (N \cup Neighbor(v)) - E$ {$N$: the set of vertices that can join to the occurrence; $E$: the set of vertices that are neighbors but cannot join to the occurrence; $v$: parent node; $Neighbor(v)$: the vertices connect to $v$}
2: for each $n \in N$ do
3:     $E \leftarrow E \cup \{n\}$
4:     for each $k = \{+, -\}$ do
5:         $z \leftarrow Join(v.interval, n.interval[k], w)$ {$z$: intervals of trends; $w$: intersection constraints}
6:         if $z \neq \emptyset$ then
7:             create a new node $v'$ for $(n, z, k)$
8:             add $v'$ to parent's ($v$) children list
9:         end if
10:        $Build(v', N, E)$
11:    end for
12: end for
```

it can intersect with any trend intervals without reducing their length.

Given this, each time being invoked, the $Build()$ procedure will find the new neighbors from the last vertex being added to the current motif occurrence (Line 1). Then, the algorithm iterates through the vertices in N and decides which of the remaining vertices can join with it (Line 2). For each vertex, we have to consider two cases, the increasing trend intervals and the decreasing trend intervals (Line 4). We compute the intersections of the intervals from the current motif occurrence with these new intervals (Implemented by $Join()$ operation, which will be discussed shortly). If a vertex with one type of trend intervals can join with current motif occurrence (the intersection set is not empty, Line 6), we will create a new node in the tree to record the vertex together with the trend intervals and we record this new node as a new child of the current motif occurrence(Line $7 - 8$). Thus, a new type of motif occurrence is being discovered and stored. We will invoke $Build()$ recursively to expand this new motif occurrence (Line 10). Note that in order to enumerate each motif occurrence only once, after we visit each vertex in the set N, we will add to the E list (Line 3). Therefore, this vertex will not be included in the motif occurrences which are being expanded later (Line 1).

Procedure $Join(z_1, z_2, w)$

```
1: $sort(z_1), sort(z_2)$; {sort each set of intervals $z_1$ and $z_2$ based on the starting time}
2: $z \leftarrow \emptyset$
3: $j \leftarrow 1$ {beginning of $z_2$}
4: for $i = 1$ to $|z_1|$ do
5:     while $(z_2[j].end < z_1[i].start + w)$ do
6:         $j \leftarrow j + 1$ {skip interval with no valid intersection}
7:     end while
8:     $l \leftarrow j$ { begin valid intersections }
9:     while $z_2[l].start \leq z_1[i].end() - w$ do
10:        $z \leftarrow z \cup intersect(z_1[i], z_2[l])$
11:        $l \leftarrow l + 1$
12:    end while
13: end for
14: return $z$
```

A key operation in the $Build()$ operation is to find the common intervals of two sets of trend intervals. Suppose we have two sets of intervals, z_1 and z_2, the naive method will simply intersect each pair of intervals, one from z_1 and another from z_2. Thus, it will take $O(|z_1| \times |z_2|)$ intersection operations. Here, we present an efficient algorithm, which in the best case only requires linear time complexity $O(|z_1| + |z_2|)$. The algorithm is illustrated in procedure $Join(z_1, z_2, w)$. Note that the parameter w is the minimal length for the resulting interval. This algorithm utilizes a simple characteristics of both sets z_1 and z_2: none of the intervals is a subset of any other intervals in the same set. Thus, if we sort each set based on the beginning time of each trend interval, then, they are sorted by their ending time as well (Line 1). With this fact, we take each trend interval in the first set z_1 and begin to make intersections on the second set z_2 only when $z_2[j].end \geq z_1[i].start + w$, which means our intersection will be at least w units long (Line $5 - 7$). Similarly, we continue making intersections on the trend from set z_1 while $z_1[l].start \leq z_1[i].end - w$ (Line $9 - 12$). We continue iteratively through the set of trends in z_1, only making intersections where appropriate in z_2 (Line 4). The correctness of this algorithm can be achieved by the following lemma.

Lemma 2 *For a given interval $z_1[i]$, for any interval $z_2[l]$, such that $z_2[l].end \geq z_1[i].start + w$ and $z_2[l].start \leq z_1[i].end() - w$, then the length of their intersect $[max(z_1[i].start, z_2[i].start), min(z_1[i].end, z_2[i].end)]$ is greater than or equal to w.*

Proof: First, we note that $z_1[i].end - z_1[i].start \geq l \geq w$ and $z_2[i].end - z_2[i].start \geq l \geq w$. Then, we have $z_1[l].end \geq z_2[l].start + w$ and $z_1[l].end \geq z_1[l].start + l \geq z_1[l].start + w$. Similarly, we have $z_2[l].end \geq z_1[l].start + w$ and $z_2[l].end \geq z_2[l].start + l \geq z_2[l].start + w$. Thus, $min(z_1[i].end, z_2[i].end) - max(z_1[i].start, z_2[i].start) \geq w$. □

3.3 Algorithm for Frequent Trend Motif Discovery

Before we set up to introduce the algorithm to find all frequent trend motifs, we will visit the *frequency* concept first. In the original Definition 2, any subset of vertices whose induced subgraphs are isomorphic to each other will be counted towards the frequency of a motif. However, a lot of them may have significant overlaps. A slightly different approach will only consider non-overlapped occurrences [16]. Here, we will allow any two occurrences share at most one vertex [26]. In other words, no edge can be shared between two occurrences for a given trend motif. Note that such a frequency concept will allow us to use the down-closure property for the motif enumeration. Given this, the major challenge in finding frequent trend motif is how to utilize the motif occurrence tree and the down-closure property to speedup the mining process.

Algorithm 3 $ExtractFrequentMotifs(Root\, r, Support\, \theta)$

```
1: $C_1 \leftarrow \emptyset; R \leftarrow \emptyset; k \leftarrow 1$
2: $Count(C_1, r)$ {count the first level}
3: while $|C_k| \neq 0$ do
4:     $C_{k+1} \leftarrow \emptyset$
5:     for each $c \in C_k$ do
6:         $c.count \leftarrow max\_independent\_set(c.motifocc\_list)$
7:         if $c.count \geq \theta$ then
8:             $R \leftarrow R \cup \{c\}$ {record the motif $c$ in resulting set $R$}
9:         end if
10:        if $c.count \geq \theta$ or $k = 1$ then
11:            for each $v \in c.motifocc\_list$ do
12:                $Count(C_{k+1}, v)$
13:            end for
14:        end if
15:    end for
16:    $k \leftarrow k + 1$
17: end while
18: return $R$
Procedure $Count(Set\, C, Node\, v)$
19: for each $v' \in v.children$ do
20:    if $v'.interval \neq \emptyset$ then
21:        $code \leftarrow canonicalcode(v')$
22:        $c \leftarrow search(C, code)$ {$c$ is created if it does not exist}
23:        $c.motifocc\_list \leftarrow c.motifocc\_list \cup \{v'\}$
24:    end if
25: end for
```

The $ExtractFrequentMotifs()$ algorithm, shown in Algorithm 3, takes the root of the motif occurrence tree r and finds all of the motifs that appear at least θ times. This is done in a level-wise fashion, similar to Apriori [3]. A key idea in this algorithm is to record each type of motif occurrence (corresponding to a node in the occurrence tree) when counting the frequency of each motif. This allows us to efficiently count the motif frequency for the next level without repeatedly accessing the same node many times. Specifically, the algorithm is as follows. It first finds the single vertex motifs by using the $Count()$ procedure on the root of the motif occurrence tree (Line 2). In $Count()$, each child of r will be visited to expand the current motif occurrence (Line 19). For each child, we create a canonical code using $canonicalcode()$ (Line 21). It searches the set C to see if any motif corresponding to this code has already been inserted. If not, it will create a new entry for this code (Line 22). Note that the $canonicalcode()$ function essentially creates a unique string for the isomorphic representation of the motif (a labeled subgraph). Many methods have been developed for such a purpose [30, 21]. Finally each motif occurrence is recored in a the $motifocc\_list$ (Line 23). After building the first level, for each set C_k, $ExtractFrequentMotifs()$ will find the maximal number of occurrences which can only overlap by no more than a single vertex for any pair of them (Implemented by *max_independent_set*, Line 6). For any level $k \geq 2$, a motif can be expanded further only if their support is at least θ (Line 10). For the case when $k = 1$, the down-closure property will not hold. Therefore, any single vertex motif will be expanded.

Table 1. Network Characteristics

Dataset	Nodes	Edges	Series	Diameter
GDP-Norm	196	375	52	8
Market82-87	116	887	250	5
Market95-00	116	607	250	6
Micro-Array	6105	8815	18	15

Note that this algorithm can be easily extended to handle other frequency count. For instance, if we count *all* the occurrences by allowing the overlap, we can simply drop Line 6 and to expand each motif even though they are infrequent (drop Line 10). In addition, we note that finding the maximal number of occurrences which can only overlap by no more than a single vertex is essentially the problem of finding the maximal independent set problem. We can build essentially a graph such that each occurrence is a vertex, and two of them are connected by an edge if they share more than a vertex. Thus, finding the frequency of the motif is equivalent to finding the maximal independent set in this graph. Since it is a well-known that this is a NP-complete problem, we simply use a heuristic to approximate the true frequency. Our heuristic is similar to the one described in [26].

4 Experimental Results

In order to find trend motifs in real networks, we tested datasets from biology, financial markets and global economics. The first data set is for the protein interaction network in the yeast *S. cerevisiae* [6], and the vertex time series is derived from mRNA microarray expression data [28]. The second and third datasets are derived from the daily market prices of 116 publicly traded companies spanning nearly twenty years from 1982 to 2000 [22, 23]. The fourth dataset is derived from the global trade and gross domestic product (GDP) data from 196 countries between the years 1948 and 2000[12]. The basic characteristics of these datasets are in Table 1 and their detailed construction is as follows. Note that all the underlying networks are undirected.

GDP-Norm This dataset is created from the publicly available Expanded Trade and GDP Data [12]. The data represents the yearly imports and exports, total trade and gross domestic product of 196 countries spanning the 52 years 1948-2000. The time series for each county is the proportion of its share in the global economy according to its gross domestic product(GDP) for that year. In other words, the time series for GDP-Norm is the normalized value of each individual annual GDP, divided by the total GDP for all countries during that year. The topology for the graph was created by comparing the yearly total trade for each country and its trade with each of the other countries. If the trade between country A and country B in any given year accounts for more than 10% of either country's total trade for that year, an edge is created between the the two countries.

Financial Stock Market The market data was split into two 5 year time spans, the first ranging from 1982 to 1987 (Market82-87), and the second from 1995 to 2000 (Market95-00). The time series for each of the 116 stocks was created by taking the log of the weekly average for each week in the time span, creating a series 250 units long. The underlying graph that correlates these stocks was created using price correlation coefficients [18]. An edge is created between two companies if those two comapnies are among the 150 highest correlated pairs from each 6-month interval.

Micro-Array The protein interaction network was constructed from high-quality multivalued data for yeast, collected from multiple databases [6]. The associated vertex time series was generated from mRNA microarray expression data on the yeast cell cycle, for which populations of yeast cells had been synchronized using α factor [28, 1]. The time series consists of 18 sample points, each 7 minutes apart, over the length of the experiment.

Output and Performance In the experiments, all trends were found with either $\sigma = 2$ or $\sigma = 3$ as the maximum time step, since a series that increases or decreases by δ at least every two or three steps can reasonably be considered as moving consistently. Additionally, the maximum depth was constant at six, ensuring that we would enumerate all occurrences of motifs that contain up to 6 vertices. In Tables 2, 5, 3, and 4 we can see the results of the experiments for each dataset at different support levels. Given different parameters σ, δ, l and w, we first show the total number of maximal intervals of increasing trends $I+$ and decreasing trends $I-$. We also show the number of vertices which have intervals of both increasing and decreasing trends, denoted as $|N+,-|$, and only have intervals of increasing trends, decreasing trends and none, denoted as $|N+|$, $|N-|$, and $|None|$. Then, we vary the support level from high to low, and report the total number of frequent trend motifs at each support level ($Count$) and the running time $Time$. Clearly, as the support level is reduced, more motifs are being discovered and the running time is increasing. However, throughout all these experiments, the running is consistently less than 3 minutes even when the support level is as low as 1.

Significant Trend Motifs For each of the datasets, many frequent trend motifs were discovered. Here, we show several representative examples from our experimental results, and list them in Tables 8, 7, and 6. Besides providing their frequency ($count$) in the corresponding datasets, we also compare them with randomized networks. Since our datasets combine both network topology and time series data, we will construct three types of randomized networks. The first type of randomization, referred to as RS, shuffles the time series data for each vertex and the underlying network topology remains the same. The second type, referred to as RN, shuffles the edges and labels (corresponding trends) among the vertices while preserving the degree distribution of each vertex [19], and the time series data remains the same. Finally, the third type of randomization, referred to as RS/RN, is a combination of the first two. We build 200 randomized networks for each type of randomization, and compute the average and standard deviation of frequencies for each trend motif in the 200 networks. Finally, we compute the Z-score for the significance of each motif as compared to the specific type of randomization.

The GDP-Norm dataset contains very interesting motifs. In the GDP-Norm motifs shown in Table 6, we see a very distinct dependence relationship among the countries. Very few motifs were found where all vertices were well connected, leading to the notion that the country with the highest degree can greatly affect its dependent neighbors. This would be further validated when we look at the specific trend motif occurrences.

In Table 7 we see two motifs that have increasing intervals on all vertices. Because we are taking the absolute value of the stock prices over five years, we can expect that the major trends in the market present the stock prices as increasing. In addition, these motifs are relatively well-connected as we expect that the related companies are likely to affect each other at a higher degree. Also, these motifs were found to be significant in both the Market82-87 network as well as the Market95-00 network. These motifs show that the underlying dynamic of the market trends is similar, regardless of the time period and that we have very similar movement between companies that are highly correlated.

Two motifs from the Micro Array dataset are shown in Table 8. These motifs display configurations of interacting

Table 2. GDP-Norm

$\delta = 0.00014, \sigma = 2, l = 10, w = 8$					
$I+$	$I-$	$\|N+,-\|$	$\|N+\|$	$\|N-\|$	$\|None\|$
48	79	18	24	48	106
Support	60	40	15	6	1
Count	66	193	301	405	1055
Time	0.01s	0.07s	1.01s	50.27s	128.7s

$\delta = 0.0002, \sigma = 3, l = 15, w = 10$					
$I+$	$I-$	$\|N+,-\|$	$\|N+\|$	$\|N-\|$	$\|None\|$
21	69	6	14	59	117
Support	40	20	10	3	1
Count	65	134	154	202	322
Time	0.01s	0.02s	0.11s	3.38s	9.10s

Table 3. Market82 Performance

$\delta = 0.019, \sigma = 2, l = 12, w = 8$					
$I+$	$I-$	$\|N+,-\|$	$\|N+\|$	$\|N-\|$	$\|None\|$
86	22	9	54	10	43
Support	60	30	15	6	1
Count	166	211	308	452	744
Time	0.1s	0.42s	3.10s	29.52s	33.44s

$\delta = 0.05, \sigma = 3, l = 12, w = 8$					
$I+$	$I-$	$\|N+,-\|$	$\|N+\|$	$\|N-\|$	$\|None\|$
74	26	10	42	12	52
Support	40	20	10	6	1
Count	226	342	427	597	892
Time	1.03s	9.86s	63.56	118.1s	138.9s

Table 4. Market95 Performance

$\delta = 0.025, \sigma = 2, l = 10, w = 6$					
$I+$	$I-$	$\|N+,-\|$	$\|N+\|$	$\|N-\|$	$\|None\|$
161	91	48	40	13	15
Support	50	30	15	6	1
Count	360	409	560	828	1173
Time	0.59s	0.64s	22.58s	57.78s	71.6s

$\delta = 0.04, \sigma = 3, l = 12, w = 12$					
$I+$	$I-$	$\|N+,-\|$	$\|N+\|$	$\|N-\|$	$\|None\|$
195	121	63	33	10	10
Support	50	30	15	6	1
Count	448	478	583	918	1202
Time	1.15s	1.52s	48.25s	160.1s	170.1s

Table 5. MicroArray

$\delta = 0.02, \sigma = 2, l = 10, w = 5$					
$I+$	$I-$	$\|N+,-\|$	$\|N+\|$	$\|N-\|$	$\|None\|$
208	633	15	190	590	5310
Support	60	30	15	6	1
Count	913	954	1001	1112	1331
Time	0.07s	0.23s	0.76s	11.15s	19.27s

$\delta = 0.06, \sigma = 3, l = 12, w = 5$					
$I+$	$I-$	$\|N+,-\|$	$\|N+\|$	$\|N-\|$	$\|None\|$
153	646	27	123	593	5362
Support	60	30	15	6	1
Count	990	1069	1216	1399	1944
Time	0.56s	2.62s	43.85s	63.67s	82.16s

Table 6. GDP-Norm Motifs

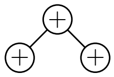
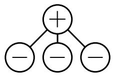
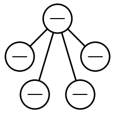
$\delta = 0.00014, \sigma = 2, l = 10, w = 8$				
Motif		RS	RN	RS/RN
	Count: 15			
	$\mu \pm \sigma$:	$.12 \pm .68$	2.30 ± 2.55	$.01 \pm .16$
	Z score:	21.9	4.99	96.6
	Count: 7			
	$\mu \pm \sigma$:	0 ± 0	$.65 \pm 1.49$	0 ± 0
	Z score:	7	4.26	7
	Count: 7			
	$\mu \pm \sigma$:	0 ± 0	1.82 ± 2.44	0 ± 0
	Z score:	7	2.12	7

Table 7. Market82-87/Market95-00 Motifs

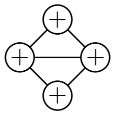
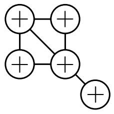
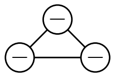
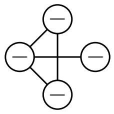
Market82: $\delta = 0.019, \sigma = 2, l = 12, w = 8$				
Market95: $\delta = 0.025, \sigma = 2, l = 10, w = 6$				
Motif		RS	RN	RS/RN
	Count: 14 (Market82), 12 (Market95)			
	$\mu \pm \sigma$:	0.09 ± 0.9	3.39 ± 2.06	$.09 \pm .54$
	Z score:	13.6	4.18	22.15
	Count: 10 (Market82), 9 (Market95)			
	$\mu \pm \sigma$:	$.06 \pm .60$	3.05 ± 2.05	$.05 \pm .50$
	Z score:	15.0	2.89	17.87

Table 8. Micro Array Motifs

$\delta = 0.02, \sigma = 2, l = 10, w = 5$				
Motif		RS	RN	RS/RN
	Count: 11			
	$\mu \pm \sigma$:	$.19 \pm .74$	$.23 \pm .50$	$.01 \pm .12$
	Z score:	14.7	21.8	91.5
	Count: 10			
	$\mu \pm \sigma$:	$.03 \pm .42$	8.91 ± 4.0	0 ± 0
	Z score:	23.9	2.52	11

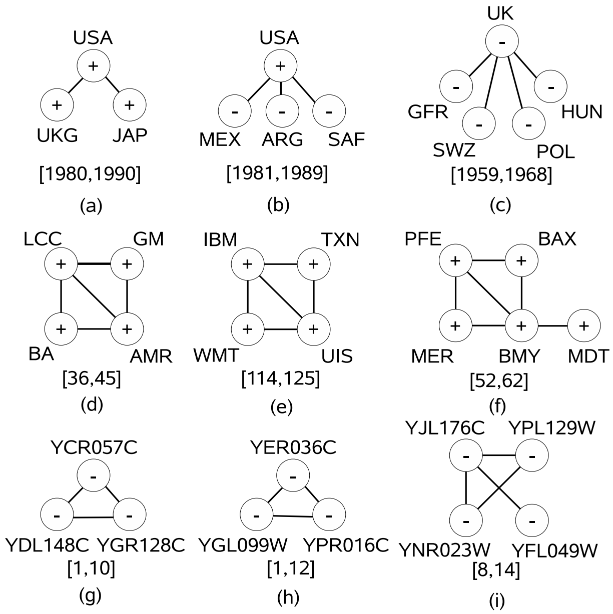


Figure 1. Example Motif Occurrences

proteins that are significantly co-regulated over longer periods of time. Note that, no vertices with increasing trends take part of trend motifs that contain a cycle. Also, for single edge motifs, increasing trend vertices appear to be underrepresented. Consequently, we hypothesize there exists an effective ”repulsion” between nodes with increasing trends. Future research will be aimed at investigating possible biological mechanisms for this effect.

Interesting Trend Motif Occurrences In Figure 1 we show some interesting trend motif occurrences that were found in each dataset. In (a), (b), and (c) we find motifs that occurred in the GDP-Norm dataset. The first motif (a), displays the partnership between the United States (USA), United Kingdom (UK), and Japan (JAP) during the 1980’s which shows significant market share growth for all three countries. In (b), however, we see that countries that depended on the United States (USA), such as Mexico (MEX), Argentina (ARG), and South Africa (SAF), were losing global market share during that same period. We believe this displays a shift in the global economic structure. Finally, in (c), we note that several regional patterns also developed as motifs. Here we see a trend where the United Kingdom (UK) is decreasing, while the European countries that depend on it, such as Germany (GFR), Switzerland (SWZ), Poland (POL), and Hungary (HUN), are also decreasing during the 60’s. Another interesting fact is that major motif occurrences found in GDP-Norm were occurring on approximately the 1955-1965 time span, and then again in the 1980 to 1990 time span. We believe that these two distinct time-based patterns can be due to the reconstruction efforts and emerging countries after World War II and then again during the waning years of the Cold War. Both eras marked major changes in the global economy and are portrayed through our identified motifs.

The second set of examples in Figure 1, are from the financial market dataset. The first, (d) from the Market82-87 dataset, displays the partnerships between US Airways Group (LCC), General Motors (GM), Boeing Company (BA), and AMR Corproation (AMR), the owner of American Airlines. The second motif in (e), from the Market95-00 dataset, shows the partnerships between three technology companies and a consumer company, namely Int’l Business Machines (IBM, computer hardware), Texas Instruments (TXN, semiconductors), Unisys Corporation (UIS, computer services) and a consumer retail company, Wal-Mart Stores, Inc (WMT). The third motif in (f), also from the Market95-00 dataset, shows the partnership between four healthcare companies and one major investment firm. These companies are Pfizer (PFE, major drugs), Baxter International Inc (BAX, medical equipment), Bristol-Myers Squibb (BMY, major drugs), Medtronic Inc (MDT, medical equipment) and finally, Merrill Lynch & Co (MER, investment services).

The third set of example motifs is taken from the yeast protein interaction network. In Figure 1 (g), the identified trend motif takes part in the small nucleolar ribonucleoprotein in yeast, which is a complex involved in the processing of rRNA found in the nucleolus of eukaryotic cells. If either of the identified genes are disrupted, the yeast cells are no longer viable. While the trend motif in (h) is isomorphic to that in (g) and its constituents are also essential for the survival of the cell, these proteins are all involved in the 60S ribosome biogenesis. The trend motif in panel (i) consists of four proteins that take part in the SWI/SNF chromatin remodeling complex that regulates transcription of many genes. In contrast to (g) and (h), these genes are not essential for the survival of the organism, however, their impairment induces multiple growth defects on the yeast cells.

We are convinced that these motifs not only are statistically significant, but they identify key characteristics about the underlying dynamics of these complex systems. The yeast motifs highlight protein complexes with important cellular functions during different parts of the cell cycle, the GDP-Norm motifs display highly correlated subgraphs that show the major shifts in global economics, while the financial market motifs display interesting partnerships between companies and their performance similarities.

5. Related Work

The ability to model and analyze dynamics on complex network has recently attracted significan research interests. An important set of problems is related to spreading phenomena on complex networks, such as epidemics and diffusion processes [7, 2]. Many studies have also focused on characterizing the topological change or cluster evolution of a system [17, 5, 24, 22]. However, the effects of time-evolution of vertex- or edge-weights have not previously been explicitly considered.

The correlation and pattern discovery of multiple time series has recently also gained a lot of attention, e.g. Sun *et al.* [29] applies tensor analysis to study co-evolving time series. Their approach is essentially a high-dimensional extension of the well-known PCA/SVD techniques. In addition, several algorithms have been developed to quickly identify strong correlations within a large number of time series [8, 25]. However, these analyses do not effectively utilize the underlying topology among the basic units of the complex system, and their time-series analysis cannot address the interplay between dynamics and systems-organization, as is captured by our motif analysis.

The problem of identifying network motifs, or frequent subgraphs, has been studied in large complex networks or collections of graphs [19, 27, 14]. The early efforts in graph mining apply heuristic algorithms to discover useful patterns from graph datasets [9, 10], and the down-closure property has been extensively applied to find frequent *induced* and/or *connected* subgraphs [16, 14, 15, 30, 13, 21]. However, these approaches only consider the topology of the graphs and therefore, will not capture dynamic effects as described in this paper.

6. Conclusions

In this paper, we have developed a data mining approach, making it possible to analyze evolving weighted complex networks. A list of new concepts and new algorithms enable the analysis from individual vertex (trend discovery), to a group of correlated vertices (trend motif occurrence), and to the common patterns of change (frequent trend motif) in a dynamic complex network. The detailed experimental study on three real datasets have demonstrated the significance of these patterns in uncovering significant events in the dynamic system, and to understand their characteristics. We hope our methodology will open a new avenue in applying motif mining to analyze the dynamics of complex systems.

References

[1] Yeast cell cycle analysis project. http://cellcycle-www.stanford.edu/.

[2] The role of the airline transportation network in the prediction and predictability of global epidemics. *Proc. Natl. Acad. Sci. USA*, pages 2015–2020, 2006.

[3] Rakesh Agrawal and Ramakrishnan Srikant. Fast algorithms for mining association rules in large databases. In *Proceedings of the 20th International Conference on Very Large Data Bases*, 1994.

[4] R. Albert and A.-L. Barabási. Statistical mechanics of complex networks. *Rev. Mod. Phys.*, 74:47, 2002.

[5] Lars Backstrom, Daniel P. Huttenlocher, Jon M. Kleinberg, and Xiangyang Lan. Group formation in large social networks: membership, growth, and evolution. In *KDD*, pages 44–54, 2006.

[6] Nizar N. Batada, Teresa Reguly, Ashton Breitkreutz, Lorrie Boucher, Bobby-Joe Breitkreutz, Laurence D. Hurst, and Mike Tyers. Stratus not altocumulus: A new view of the yeast protein interaction network. *PLoS Biology*, 4:1720, 2006.

[7] Stefan Bornholdt and Heinz Georg Schuster, editors. *Handbook of Graphs and Networks: From the Genome to the Internet.* Wiley-VCH, 2002.

[8] Richard Cole, Dennis Shasha, and Xiaojian Zhao. Fast window correlations over uncooperative time series. In *KDD*, pages 743–749, 2005.

[9] Diane J. Cook and Lawrence B. Holder. Substructure discovery using minimum description length and background knowledge. *Journal of Artificial Intelligence Research*, 1:231–255, 1994.

[10] L. Dehaspe, H. Toivonen, and R. D. King. Finding frequent substructures in chemical compounds. In R. Agrawal, P. Stolorz, and G. Piatetsky-Shapiro, editors, *4th International Conference on Knowledge Discovery and Data Mining*, pages 30–36. AAAI Press., 1998.

[11] Reinhard Diestel. *Graph Theory.* Springer-Verlag, 2000.

[12] Kristian S. Gleditsch. Expanded trade and gdp data,. *J. Conf. Res.*, 46:712–724, 2002.

[13] Jun Huan, Wei Wang, Deepak Bandyopadhyay, Jack Snoeyink, Jan Prins, and Alexander Tropsha. Mining protein family-specific residue packing patterns from protein structure graphs. In *Eighth International Conference on Research in Computational Molecular Biology (RECOMB)*, pages 308–315, 2004.

[14] Akihiro Inokuchi, Takashi Washio, and Hiroshi Motoda. Complete mining of frequent patterns from graphs: Mining graph data. *Mach. Learn.*, 50(3):321–354, 2003.

[15] Michihiro Kuramochi and George Karypis. Frequent subgraph discovery. In *ICDM '01: Proceedings of the 2001 IEEE International Conference on Data Mining*, pages 313–320, 2001.

[16] Michihiro Kuramochi and George Karypis. Finding frequent patterns in a large sparse graph. In *SDM*, 2004.

[17] Jure Leskovec, Jon M. Kleinberg, and Christos Faloutsos. Graphs over time: densification laws, shrinking diameters and possible explanations. In *KDD*, pages 177–187, 2005.

[18] R. N. Mantegna. Computer physics communications 121, 1999.

[19] R. Milo, S. Shen-Orr, S. Itzkovitz, N. Kashtan, D. Chklovskii, and U. Alon. Network motifs: Simple building blocks of complex networks. *Science*, 298(5594):824827, October 2002.

[20] Ron Milo, Shalev Itzkovitz, Nadav Kashtan, Reuven Levitt, Shai Shen-Orr, Inbal Ayzenshtat, Michal Sheffer, and Uri Alon. Superfamilies of evolved and designed networks. *Science*, 303:1538 – 1542, 2004.

[21] Siegfried Nijssen and Joost N. Kok. A quickstart in frequent structure mining can make a difference. In *KDD*, pages 647–652, 2004.

[22] J.-P. Onnela, A. Chakraborti, K. Kaski, and J. Kertesz. Dynamic asset trees and portfolio analysis. *Eur. Phys. J. B*, 30(3):285, 2002.

[23] J.-P. Onnela, A. Chakraborti, K. Kaski, and J. Kertesz. Dynamic asset trees and black monday. *Physica A*, 324:247, 2003.

[24] Gergely Palla, Albert-Laszlo Barabasi, and Tamas Vicsek. Quantifying social group evolution. *Nature*, 446(7136):664–667, April 2007.

[25] Spiros Papadimitriou, Jimeng Sun, and Philip S. Yu. Local correlation tracking in time series. In *ICDM*, pages 456–465, 2006.

[26] F. Schreiber and H. Schwbbermeyer. Towards motif detection in networks: frequency concepts and flexible search. In *Proc. Intl. Wsh. Network Tools and Applications in Biology (NETTAB'04)*, pages 91–102., 2004.

[27] SS Shen-Orr, R Milo, S Mangan, and U Alon. Network motifs in the transcriptional regulation network of escherichia coli. *Nat Genet*, 31:64–68, 2002.

[28] Paul T. Spellman, Gavin Sherlock, Michael Q. Zhang, Vishwanath R. Iyer, Kirk Anders, Michael B. Eisen, Patrick O. Brown, David Botstein, and Bruce Futcher. Comprehensive identification of cell cycle-regulated genes of the yeast saccharomyces cerevisiae by microarray hybridization. *Mol. Biol. Cell*, 9:3273–3297.

[29] Jimeng Sun, Dacheng Tao, and Christos Faloutsos. Beyond streams and graphs: dynamic tensor analysis. In *KDD '06: Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 374–383, 2006.

[30] Xifeng Yan and Jiawei Han. gspan: Graph-based substructure pattern mining. In *ICDM '02: Proceedings of the 2002 IEEE International Conference on Data Mining (ICDM'02)*, page 721, 2002.