Temporal and Structural Analysis of Biological Networks in Combination with Microarray Data

Chang Hun You, Lawrence B. Holder and Diane J. Cook

Abstract— **We introduce a graph-based relational learning approach using graph-rewriting rules for temporal and structural analysis of biological networks changing over time. The analysis of dynamic biological networks is necessary to understand life at the system-level, because biological networks continuously change their structures and properties, while an organism performs various biological activities. A dynamic graph represents dynamic properties as well as structural properties of biological networks. Microarray data can reflect dynamic properties of biological processes. Biological networks, which contain various molecules and relationships between molecules, show structural properties representing various relationships between entities. Most current graph-based data mining approaches overlook dynamic features of biological networks, because they are focused on only static graphs. Most approaches for analysis of microarray data disregard structural properties on biological systems. But our dynamic graph-based relational learning approach describes how the graphs temporally and structurally change over time in the dynamic graph representing biological networks in combination with microarray data.**

I. Introduction

Analysis of biological networks is one of the key ways to understand biosystems. Our bodies are not only well-organized biological networks but also dynamic systems. Biological networks include various molecules and relationships between molecules. Furthermore, the networks dynamically change their structures and properties, while organisms carry out various biological activities, such as digestion, respiration and so on. While there are many other aspects to the activity of biological networks, our focus is on the temporal and structural analysis.

A graph is a relational data structure representing data using vertices and edges, and is a natural way to represent biological networks, where vertices denote biomolecules and edges denote relations between molecules. Graph-based data mining is a process to discover novel knowledge in data represented as a graph. Several graph-based data mining approaches have been applied to identify interesting patterns in biological networks. However, the current graph-based data mining approaches overlook dynamic features of biological networks, because most of them are focused on only static graphs. Temporal data mining can mine dynamic features in the temporal sequence of biological networks. But it is hard for temporal data mining to discover structural features as well as dynamic features in the biological networks.

Chang Hun You, Lawrence B. Holder and Diane J. Cook are with School of Electrical Engineering & Computer Science, Washington State University, Box 642752, Pullman, WA 99164-2752, (email: {changhun, holder, cook}@eecs.wsu.edu})

This research proposes a novel algorithm to discover structural features along with temporal features in the temporal sequence of biological networks. Our dynamic graph-based relational learning approach uses graph-rewriting rules to analyze how biological networks change over time.

Graph-rewriting rules define how one graph changes to another in its topology replacing vertices, edges or subgraphs according to the rewriting rules. First, we generate a dynamic graph, which is a sequence of graphs representing biological networks changing over time. Then, our approach discovers rewriting rules, which show how to replace subgraphs, between two sequential graphs. Discovered graph rewriting rules give us two aspects of novel knowledge. First, temporal patterns in graph rewriting rules show how the graphs change over time, such as periodic repeating of graph rewriting rules or temporal orders among several rules. Second, graph rewriting rules describe how the substructures in rules connect to the parent graph, in other words, how the molecules are related to the biological network at the specific time. The graph rewriting rules learned by our approach can describe how the structures of graphs change and how substructures in rules are related to other substructures in graphs. This approach enables us to investigate dynamic patterns in biological networks for both aspects: temporal and structural analyses.

First, we discuss several works related to structural and temporal analysis of biological networks including microarray analysis and several computational methods. We also define the graph rewriting rules for our research. We present our Dynamic Graph Relational Learning (DynGRL) algorithm with several experiments. In our experiments we generate dynamic graphs of the citrate cycle metabolic pathways and MAPK pathways using KEGG PATHWAY database and microarray data of yeast. Then, we apply our DynGRL approach to the dynamic graphs. The results show our discovered graph rewriting rules and temporal patterns in rewriting rules such as periodic repeating and temporal orders.

The goal of this research is, first, to discover novel temporal patterns in the graph rewriting rules to describe structural changes of graphs in a dynamic graph. The second is to visualize and understand how the biological networks change their structures. The next step would be to automate the general rule discovery phase.

II. Related Works

A. Study of Biosystems

Analysis of biological networks is an important area in systems biology. Bioinformatics has been focused on

molecular-level research until now. Genomics and proteomics, main areas in molecular-level research, have studied function and structure of macromolecules in organisms, and produced a huge amount of results. However, there are few molecules (i.e., DNA, RNA, protein, and so on) that can work alone. Each molecule has its own properties and relationships with other molecules to carry out its function. Biological networks have various molecules and relations between them including reactions and enzyme-relations among genes and proteins. Here, we define the structure as the relation between bio-molecules. In addition to the structural aspect, we also consider the temporal aspect of biological networks, because the biosystems always change their properties and structures while interacting with other conditions.

There is much research in the study of biological networks. Mathematical modeling, which is an abstract model to describe a system using mathematical formulae [1], is a well-known approach to biological networks. Most of these approaches, as a type of quantitative analysis, model several kinetics of pathways and analyze the trends in the amount of molecules and flux of biochemical reactions. But this approach often oversimplifies pathways and disregards structural aspects of biological networks like relations among multiple molecules.

The microarray is another approach to study biosystems. The microarray is a tool for monitoring gene expression levels for thousands of genes at the same time [2], [3]. Microarrays can take a snapshot of gene expression levels for a large amount of genes for each condition in an experiment and have already produced terabytes of important functional genomics data that can provide clues about how genes and gene products interact and form their gene interaction networks. Most genes are co-expressed as most proteins interact with other molecules. Co-expressed genes can represent common processes or patterns in biological networks (gene regulatory networks or protein networks) in the specific condition. Patterns in gene expression levels can describe changes in the biological status or distinguish two different states, such as the normal and disease state. But the microarray analysis can overlook structural aspects, which show how the genes or expressed gene products are related to each other in biological networks.

It is necessary to analyze biological networks not only for the structural aspect but also for the temporal aspect for a system-level understanding of organisms.

B. Computational Approaches

Graph-based data mining is to discover novel knowledge in graph-represented data. Graph-based data mining [4], [5] has been successfully applied to biological networks. This approach represents biological networks as graphs, where vertices represent molecules and edges represent relations between molecules. Graph-based data mining discovers frequent structural patterns in biological networks, but overlooks temporal properties.

Temporal data mining is to mine temporal patterns in sequential data, which is ordered with respect to some index like time stamps, rather than static data [6]. Temporal data mining focuses on discovery of relational aspects in data such as discovery of temporal relations or cause-effect association. In other words, we can understand how or why the object changes rather than merely static properties of the object.

There are several researches to apply temporal data mining in biological data. Ho et al. [7] propose an approach to detect temporal patterns and relations between medical events of Hepatitis data. Farach-Colton et al. [8] introduce an approach of mining temporal relations in protein-protein interactions. They model the assembly pathways of Ribosome using protein-protein interactions. Temporal data mining approaches discover temporal patterns in data, but they disregard relational aspects among entities. For example, they can identify temporal patterns of appearance of genes such that a gene, YBR218C, appears before an other gene, YGL062W, but cannot identify how these two genes interact with each other.

There are many aspects to consider for understanding biological networks, but our research focus on two aspects. First, we need to focus on relationships between molecules as well as a single molecule. Second, we should consider biological networks as dynamic operations rather than static structures, because every biological process changes over time and interacts with inner or outer conditions. It is necessary to analyze biological networks not only for structural aspects but also for dynamic aspects for a system-level understanding of organisms. For this reason, we need an approach to analyze biological networks changing over time in both aspects: structural and dynamic properties. There are many other factors (e.g., concentrations, regulatory feedback) that affect the behavior of biological networks but are not represented in the KEGG data. In future work, we will introduce these factors into our representation and learn rules based on them as well.

III. Graph Rewriting Rules

Traditional graph grammar approaches determine which subgraphs will be replaced by other subgraphs. We focus on representing changing structures, and ultimately we plan to learn general rules in the discovered rewriting rules.

Graph rewriting is a method to represent structural changes of graphs using graph rewriting rules [9], [10]. Generally, graph rewriting rules identify subgraphs in a graph and modify them. Each graph rewriting rule defines a transformation between L and R, where L and R are subgraphs in two graphs G and H respectively, such that L is replaced by R, L is deleted, or R is created [11].

We define our graph rewriting rules to represent how substructures change between two graphs rather than just what subgraphs change. First, we discover maximum common subgraphs between two sequential graphs G_1 and G_2. Then, we derive removal substructures from G_1 and addition substructures from G_2. Figure 1 shows an instance of this process. A maximum common subgraph (denoted by S) is discovered between two graphs, G_1 and G_2. Then the remains in G_1 and G_2 become removal (denoted by

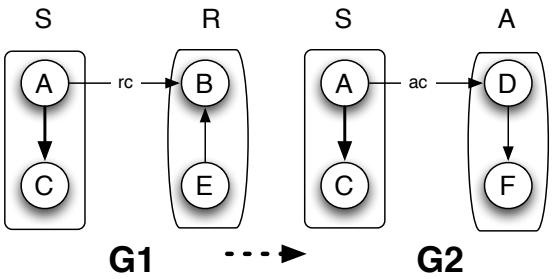

Fig. 1. An example of application of graph rewriting rules, where S denotes the maximum common subgraph between two graphs G_1 and G_2, R denotes removal subgraphs and A denotes addition subgraphs. rc and ac denote the connection edges for the removal and addition rules.

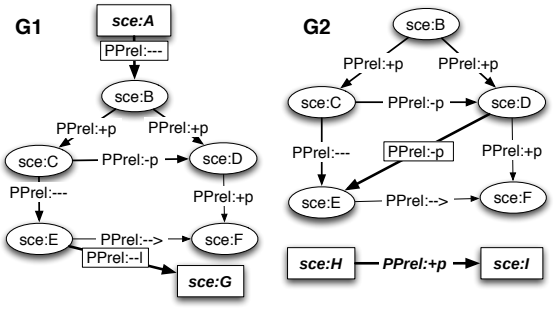

Fig. 2. An instance of graph rewriting rules in synthetic biological networks. The subgraphs containing the ellipse-shape vertices denote common substructures in G_1 and G_2. The subgraphs containing the rectangle-shape vertices denote the removal (in G_1) and addition (in G_2) substructures. The box-labeled edges denote the connection edges.

R) and addition (denoted by A) substructures respectively. These substructures with connection edges rc and ac are elements of graph rewriting rules: removal and addition rules respectively. Here, we define several preliminary terms.

A labeled and directed graph G is defined as $G = (V, E)$, where V is a set of vertices and E is a set of edges. An edge e ($\in E$) is directed from x to y as $e = (x, y)$, where $x, y \in V$. The graph represents a biological network, where vertices are labeled by names of entities (molecules, reactions, and relations), and edges are labeled by names of relationships between two entities. Each label represents the identification number or attribute from the KEGG data [12]. The dynamic graphs are essentially subgraphs of the KEGG network, where an edge in the KEGG network is also in the dynamic graph if and only if the microarray activation levels of the two entities (genes) connected by that edge exceed a certain threshold.

Now, we define a dynamic graph DG as a sequence of n graphs as follows.

$$DG = \{G_1, G_2, \cdots, G_n\}$$

Each graph G_i is a graph at time i for $1 \leq i \leq n$. Then, we define a set of removal substructures RG and a set of addition substructures AG as follows.

$$RG_i = G_i / S_{i,i+1}$$
$$AG_{i+1} = G_{i+1} / S_{i,i+1}$$

RG_i denotes a set of removal substructures in a parent graph G_i, AG_{i+1} denotes a set of addition substructures in a parent graph G_{i+1}, and $S_{i,i+1}$ is a maximum set of common subgraphs between two sequential graphs G_i and G_{i+1} in a dynamic graph DG.

A prior graph G_i is transformed to a posterior graph G_{i+1} by application of a set of graph rewriting rules $GR_{i,i+1}$ as follows.

$$G_{i+1} = G_i \bigoplus GR_{i,i+1}$$

A set of graph rewriting rules $GR_{i,i+1}$ between two sequential graphs G_i and G_{i+1} is defined in combination with RG, AG, CE and CL as follows.

$$GR_{i,i+1} = \{(m, p, CE_m, CL_m), \cdots, (n, q, CE_n, CL_n), , \cdots, \}$$

m and n are indices of graph rewriting rules in a set $GR_{i,i+1}$. p and q are indices of a removal substructure in RG_i and an addition substructure in AG_{i+1} respectively. CE and CL are defined as a set of connection edges and a set of labels of the connection edges. Each element of RG and AG corresponds to a set of CE and CL, unless a removal (addition) substructure does not connect to the parent graph. CE_k and CL_k represent connections between substructures and parent graphs ($k = m$ or n) as follows.

$$CE = \{(d, X, Y), \cdots\},$$
$$CL = \{label_{xy}, \cdots\}$$

d represents whether the edge is directed or undirected using d and u. X and Y denote vertices as a starting and ending of the edge. Because the connection edge links the substructure to the parent graph, one end of this edge is from the substructure and the other is from the parent graph. The end vertex from the substructure starts with "s" followed by the index of the vertex, and the end vertex from the parent graph starts with "g" followed by the index of the vertex. For example, $(d, g1, s3)$ represents the directed edge from a vertex 1 in the parent graph to another vertex 3 in the substructure. $label_{xy}$ represents a label for the corresponding connection edge between two vertices X and Y. The number of elements of CE (CL as well) represents the number of connections between substructures and the parent graph. If a substructure is not connected with the parent graph, both sets of CE and CL are empty. We will describe more detailed examples in the results section.

Figure 2 shows an example of graph rewriting rules between two graphs, G_1 and G_2 in synthetic biological networks. The subgraphs containing the ellipse-shape vertices in both graphs represent common substructures. The rectangle-shape vertices elements in G_1 represent removal substructures (from G_1) and the rectangle-shape elements in

G_2 represent addition substructures (to G_2).

$$\begin{aligned} GR_{1,2} \quad = \quad & \{(r_1, rSub_1, \{(d, s1, g2)\}, \{PPrel : - - -\}), \\ & (r_2, rSub_2, \{(d, g5, s1)\}, \{PPrel : - - |\}), \\ & (a_1, \emptyset, \{(d, g3, g4)\}, \{PPrel : -p\}), \\ & (a_2, aSub_1, \emptyset, \emptyset)\} \end{aligned}$$

$GR_{1,2}$ represents a set of graph rewriting rules, which is applied to G_1 and produces G_2 using $G_2 = G_1 \bigoplus GR_{1,2}$ as described previously. It has three graph rewriting rules. For example, r_1 (r denotes removal.) represents an index of removal rules including a removal subgraph ($rSub_1$), which contains a single vertex *sce:A*. $rSub_1$ was connected by an edge $(d, s1, g2)$, which is labeled by $PPrel : - - -$. This edge is a directed edge (indicated by 'd'). This edge is from $s1$, which denotes a vertex number 1 in $rSub_1$ (s denotes the substructure) to $g2$, which denotes a vertex number 2 in G_1 (g denotes the parent graph). r_2 represents the removal rule including the single-vertex substructure labeled as *sce:F* and the connection edge $PPrel : - - |$. a_1 and a_2 represent addition rules similarly. a_1 has empty as the additional substructure, because a_1 is a rule representing an edge with the italic font label $PPrel : -p$ in G_2 without any addition substructure. a_2 has empty for edges and edge labels, because $aSub_1$ represents a disconnected graph including vertices *sce:H* and *sce:I* in G_2.

The graph rewriting rules show how two sequential graphs are structurally different. After collecting all sets of graph rewriting rules in a dynamic graph, we also discover temporal patterns in graph rewriting rules, which can describe how the graphs change over time as well as which structures change.

IV. Dynamic Graph-Based Relational Learning

The first goal of this research is to discover graph rewriting rules in a dynamic graph representing biological networks changing over time. This section describes our Dynamic Graph-Based Relational Learning (DynGRL) approach to discover graph rewriting rules in a dynamic graph.

A. Algorithm

The algorithm starts with a dynamic graph DG consisting of a sequence of n graphs as shown in algorithm 1. First, the algorithm creates a list of n virtual graphs, VGL, corresponding to n time series of graphs at line 1. Our approach uses a virtual graph to specify the applying locations of graph rewriting rules. Because a graph may have multiple graph rewriting rules and several same-labeled vertices and edges, the exact locations of connections edges and rewriting rules are important to reduce the discovery error. The next procedure is to create a two-graph-set, $Graphs$, including two sequential graphs G_i and G_{i+1} (line 5) and to specify the $limit$ based on unique labeled vertices and edges of G_i and G_{i+1} (line 6). UVL and UEL denote the number of unique vertex labels and edges in G_i and G_{i+1}. The $Limit$ based on the number of labels in the input graph bounds the search space within polynomial time and ensure consideration of most of the possible substructures.

An inner loop (line 7 to 14) represents procedures to discover common substructures between two sequential graphs. We use the SUBDUE graph-based relational learning approach to discover the maximum common substructures [13], [14]. SUBDUE evaluates substructures using the Minimum Description Length (MDL) principle to find the best substructure which minimizes the description length of the input graph after being compressed by the substructure. More detail on the evaluation approach is described in [13]. The maximum common subgraph is a reasonable basis from which to identify changes, and the best MDL substructure finds this subgraph or something close to it most of the time, but in a polynomial amount of time. Even though to find the maximum common subgraph is NP-Complete, SUBDUE can be used as a polynomial-time approximation to this problem using $Limit$ and iteration as described later in the next section.

Algorithm 1: Discovery algorithm

Input: $DG = \{G_1, G_2, \cdots, G_n\}$
Output: RRL

1 Create $VGL = \{VG_1, VG_2, \cdots, VG_n\}$
2 $RRL = \{\}$
3 **for** $i = 1$ *to* $n - 1$ **do**
4 $RemSubSet$ = $AddSubSet$ = $ComSubSet$ = $\{\}$
5 $Graphs = \{G_i, G_{i+1}\}$
6 $Limit = UVL + 4(UEL - 1)$
7 **while** *No more compression* **do**
8 BestSub = DiscoverSub(Limit, Graphs)
9 **if** $BestSub \in G_i$ & G_{i+1} **then**
10 Add $BestSub$ into $ComSubSet$
11 **end**
12 Compress $Graphs$ by $BestSub$
13 Mark $BestSub$ on VG_i and VG_{i+1}
14 **end**
15 Get $remSubs$, CE from VG_i
16 Add $remSubs$ into $RemSubSet$ and CE into $RemCESet$
17 Get $addSubs$, CE from VG_{i+1}
18 Add $addSubs$ into $AddSubSet$ and CE into $AddCESet$
19 Create RR from $RemSubSet$, $AddSubSet$, $RemCESet$, $AddCESet$
20 Add RR into RRL
21 **end**

After discovery of the best substructure, the algorithm checks whether the substructure is a subgraph of both graphs G_i and G_{i+1}. In the affirmative case, the best substructure is added into $ComSubSet$ and the two target graphs are compressed by replacing the substructure with a vertex. If the best substructure does not belong to one of the two graphs, the algorithm just compresses the graphs without adding any entry into $ComSubSet$. After compression, the algorithm discovers another substructure at the next iteration until there is no more compression.

Using the complete list of common substructures, $ComSubSet$, the algorithm acquires removal substructures, $remSubs$, and addition substructures, $addSubs$, (line 15 and 17). First, the algorithm identifies vertices and edges not part of common substructures and finds each disconnected

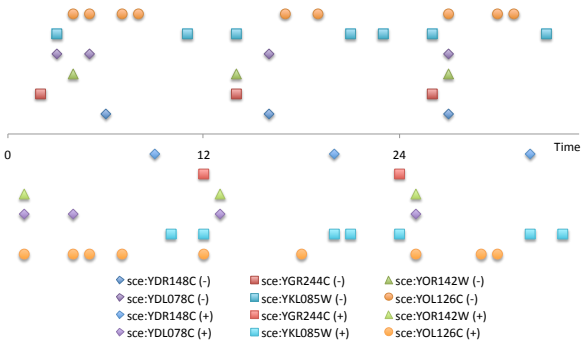

Fig. 3. A visualization of time points when the substructure including each gene is removed from or added to graphs representing the citrate cycle pathway at the experiment of threshold 0.6. Genes with (-) represent removal time points and genes with (+) represent addition time points.

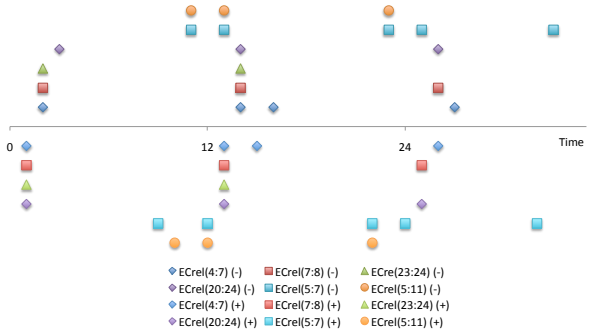

Fig. 4. A visualization of time points when a particular substructure is removed from or added to graphs representing the citrate cycle pathway at the experiment of threshold 0.6. Each substructure includes a relation, which is an enzyme-enzyme relation between two gene, where $ECrel(x, y)$ represents the relation, and x, y represent the enzyme vertices.

substructures in G_i and G_{i+1} using the modified Breadth First Search (mBFS), which adds each edge as well as each vertex into the queues as visited or to be visited. The identified substructures in G_i and G_{i+1} are removal and addition substructures respectively. While mBFS searches these removal and addition substructures, it also finds connection edges, CE, as described previously. These edges are added into $RemCESet$ and $AddCESet$, where removal and addition substructures are added into $RemSubSet$ and $AddSubSet$ respectively (in line 16 and 18). Using these rewriting substructures and connection edges, rewriting rules are created (in line 19 to 20).

B. Current Challenges

The main challenge of our algorithm is to discover maximum common subgraphs between two sequential graphs. The maximum common subgraph problem is known to be NP-hard [15]. We try to avoid this problem, first, using the $limit$ parameter to restrict the number of substructures to consider in each iteration. Second, our algorithm does not try to discover the whole common substructures at once. In each step, the algorithm discovers the small portion of common, connected substructures and iterates the discovery process until discovering the whole maximum common subgraphs. Usually, the size of graphs representing biological networks changing over time is not very large. Therefore, discovery of graph rewriting rules is still feasible. However, we still have challenges to analyze very large graphs.

V. Application and Results

We prepare dynamic graphs representing the citrate cycle metabolic pathway and MAPK pathway in combination with a microarray data. Then, our graph rewriting rule discovery system, DynGRL, discovers graph rewriting rules in the dynamic graphs. Our results show several temporal patterns of graph rewriting rules and structural changes in the pathways.

A. Microarray Data and Graph Representation

Tu et al. [16] observe periodical gene expression of *Saccharomyces cerevisiae* using microarray analysis. They use 36 sets of microarray data for every 25 minutes. The results show more than 50% of genes (usually involved in metabolism) have three periodic cycles in the gene expression. Here, we generate dynamic graphs that contains 36 graphs representing the citrate cycle metabolic pathways and MAPK pathways in combination with the microarray data. 30 genes out of 7,188 are shown in the citrate cycle pathway and 55 genes are shown in the MAPK pathway. We normalize each gene expression value from 0 to 1, because we are focused on trends of the changes of gene expression values.

The TCA cycle is a series of enzyme-catalysed chemical reactions starting with the results from the glycolysis pathway and pyruvate oxidation. This pathway, also called citrate cycle, is a hub in metabolism for three reasons [17]. First, it is the most important pathway to generate ATP, an energy molecule, in aerobic organisms. Second, this pathway provides intermediates for many other pathways. Third, the TCA cycle is closely regulated in coordination with other pathways. The MAPK (Mitogen-activated protein kinase) pathway regulates various processes such as proliferation, cell mating, cell division, and apoptosis [18]. A kinase (also called phosphotransferase) is an enzyme to catalyze the phosphorylation process which transfers phosphate groups (PO_4^{3-}) from one molecule to the other. The MAPK pathway phosphorylates various proteins including transcription factors and cytoskeletal proteins to influence other pathways.

First, we generate static graphs to represent sce00020 (the citrate cycle pathway) and sce04010 (the MAPK pathway) of yeast (*Saccharomyces cerevisiae*) from the KEGG PATHWAY database [12], where vertices represent compounds, genes, enzymes, relations and reactions, and edges represent relationships between vertices. Here, we assume only genes are changed over time, and the amount of other molecules like chemical compounds constantly remain the same. We use a threshold t for applying numeric gene expression values on graph. At each time, we assume a gene, which has more than t gene expression value, is shown in the graph.

One particular point is our graph representation has enzyme vertices, which do not exist in KEGG data. KEGG

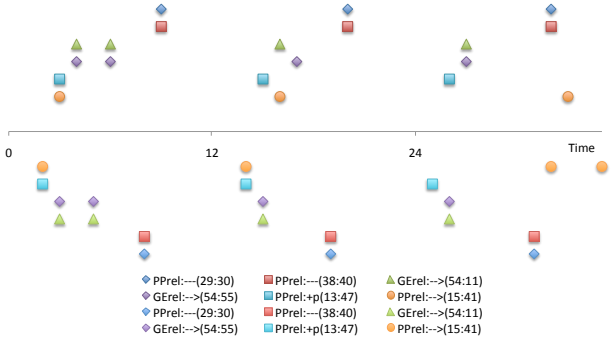

Fig. 5. A visualization of time points when the substructure including each relation is removed from or added to graphs representing the MAP kinase pathway at the experiment of threshold 0.6.

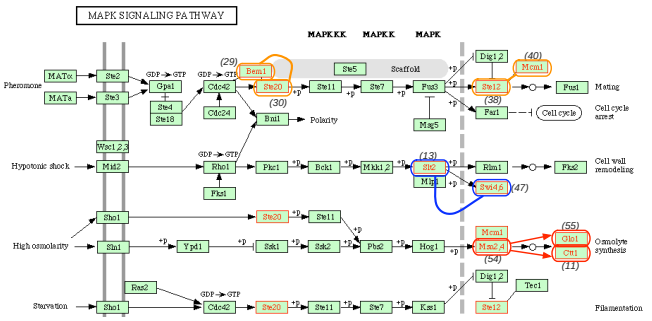

Fig. 6. A visualization of relations on the MAPK pathway downloaded from [12]. The number, (a), denotes the number x, y in $PPrel(x, y)$ and $GErel(x, y)$. The blue marked relation ($PPrel(13, 47)$) is first removed or added. Then, the red marked relations ($GErel(54, 55)$ and $GErel(54, 11)$ are removed or added. The orange marked relations ($PPrel(38, 40)$ and $PPrel(29, 30)$) are removed or added at last.

data presents a gene when they need to denote a protein or enzyme in the pathway for the specific species, because the gene generates the protein or enzyme and the gene is unique in each species. But our representation presents the enzyme vertices linked to the genes by *G_to_E* edges for representing the following scheme as well as the central dogma. If there is an protein made by two genes, A and B, our graph presents one enzyme vertex linked to two genes. At a specific time, only gene A can be expressed, but not B. Then, the enzyme cannot be generated, because the enzyme needs two genes for synthesis. Only when all genes are expressed, the enzyme vertex is shown in the graph. In that time, the reaction, which is catalyzed by the enzyme, is also shown. In this way, we can observe structures of biological networks based on the microarray gene expression at each time. Figure 7 shows examples of our graph representation. G_2 have four enzyme vertices, where each vertex has one or more links to genes.

B. Results

This section shows the temporal and structural patterns of discovered graph rewriting rules in our experiments.

1) Temporal patterns: As described previously, the goal of this research is to discover temporal patterns in graph rewriting rules to describe structural changes of biological networks changing over time. Because the result of the microarray data [16] represents three cycles of gene expression, we observe similar temporal patterns in graph rewriting rules. First, we represent the result of the citrate cycle pathway. Then, the result of the MAPK pathway follows.

Figure 3 shows the temporal patterns in removal and addition rules. This result shows the rewriting rules including 6 out of 30 genes in the TCA cycle experiment using the threshold 0.6. The points denote time points when the substructures including each gene are removed or added. The points above the time axis represent the time points when the substructures including the specified genes or relation are removed. The points below the time axis represent the time points when the substructures including the specified genes or relation are added. For example, a substructure including YGR142W genes are added and removed three times. Half of 30 genes are removed and added periodically showing three cycles. Some time points cannot easily be divided into the three cycles like YKL085W.

We focuses on graph rewriting rules including relations as well as genes. The KEGG PATHWAY database has three types of relations such as enzyme-enzyme relation, protein-protein relation and gene expression relation [12]. They are donoted as ECrel, PPrel and GErel vertices respectively. In our graph representation, a relation between genes is shown as a relation between enzymes, which are linked to the genes. A relation is shown in the graph, only when all of the related enzymes are shown. The enzymes are shown, only when all of the linked genes are shown, in other worlds, when all of the linked genes have more than t gene expression values.

Figure 4 shows the temporal patterns in enzyme-enzyme relations ($ECrel(x, y)$), where x and y denote the enzyme vertices in the graph. All six relations shown in the experiment with the threshold 0.6 are removed and added in the three periodic cycles. The periodic temporal patterns in the relations (figure 4) are observed to be more distinguishable than the temporal patterns in the genes (figure 3), because of the following reason. Suppose we have one relation between two genes, A and B, and A is always shown and B is shown for only three times. Even though the gene A is always shown in the graph, the relation is shown only when the gene B is shown for activating the enzyme-enzyme relation.

We also apply our approach to the MAPK pathway. Unlike the metabolic pathway case, there are only a few number of relations showing cycles in the MAPK pathway. Tu et al. describe most of periodically oscillated genes are involved in metabolism. In our experiment with the threshold 0.6, DynGRL discovers 20 relations including protein-protein relations (PPrel) and gene-enzyme relations (GErel). We observe only 6 out of 20 relations show cyclic removal and additions. In addition to the three cycles, we identify the different temporal patterns from the TCA cycle case. Figure 5 shows the temporal patterns discovered in the MAPK pathway. The temporal patterns show the ordered patterns as well as three cycles. The removal and addition rules show

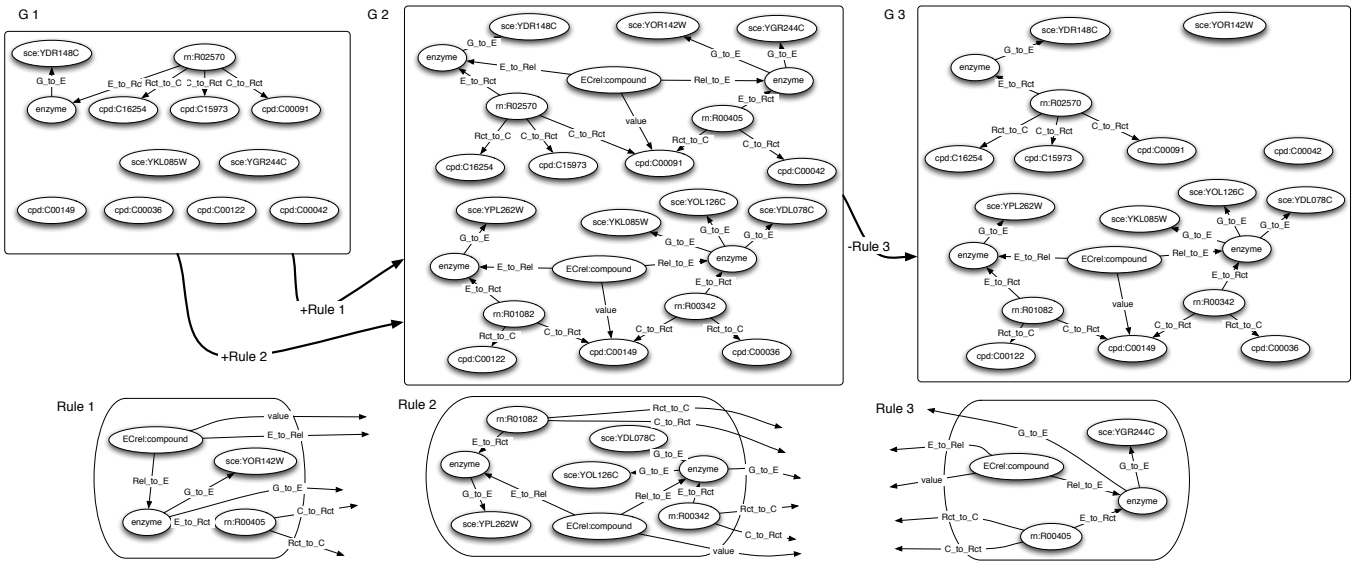

Fig. 7. Structural changes of a dynamic graph representing the partial TCA cycle. G_i graphs represent the graphs at time i. Rule 1 to 3 represent the rewriting rules used in the graph transformation from G_1 to G_3. The edges without the destination vertex in Rule 1 to 3 represent the connection edges.

an order such that, first, the relations ($PPrel(13, 47)$) are removed, then two gene-enzyme relations ($GErel(54, 55)$ and $GErel(54, 11)$) are removed, and two protein-protein relations ($PPrel(38, 40)$ and $PPrel(29, 30)$) are removed at last. They are added in the same order, too. We can also observe the additions always precede the removals, unless the graphs already contain the substructures. Figure 6 shows the ordered patterns on the MAPK pathway.

Even though we need to explore the biological meaning of the ordered temporal patterns, we can still claim our algorithm is useful. The genes and proteins in these three relation groups do not have any specific common function except that they belong to the same pathway. However, we can identify the temporal relations (ordered patterns) of the groups of relations and related genes. These patterns are hardly discovered in a static graph-based analysis. We can also address the different rate of temporal changes using the different gaps between two rules, i.e., the shorter gap would represent a faster process rather than the longer gap. This challenge is left for our future works.

In this experiment, we have shown that DynGRL discovers graph rewriting rules from a dynamic graph representing the TCA cycle pathway and MAPK pathway changing over time. These graph rewriting rules represent temporal patterns that describe how the structure of the pathways change over time by showing which elements change periodically or in order. These temporal patterns and graph rewriting rules help us to understand dynamic properties of biological networks.

2) Structural patterns: The other goal of this research is to show structural patterns in pathways. Because an advantage of the graph representation is visualization, we can understand biological networks better by representing structural changes over time. This section describes an instance of discovered substructures with graph rewriting rules.

Figure 7 shows structural changes of a dynamic graph representing the partial TCA cycle including 7 genes and two enzyme-enzyme relations. G_2 shows the graph containing all possible genes and relations. G_i represents the graph at time i. The dynamic graph in this example contains three graphs from time 1 to 3. The edges between two sequential graphs represent the graph transformation using removal (-) or addition (+) rules out of the three rules (rule 1 to 3). For example, graph G1 is transformed to G2 with two addition rules, + *rule 1* and + *rule 2*. The edges without one end vertex represent the connection edges between substructures in the rewriting rules and the parent graphs (G_i) as described previously. The connection edges describe how the discovered substructures link to the parent graph.

We also show the graph rewriting rule between two graphs as a formula. For instance, $GR_{1,2}$ is shown as,

$$\begin{aligned} GR_{1,2} &= \{(a_1, Rule_1, CE1, CL1), \\ &= \{(a_2, Rule_2, CE2, CL2\} \end{aligned}$$

where a_1 and a_2 denote the indices of addition rules, and $Rule_1$ and $Rule_2$ denote rule 1 and 2 in figure 7 respectively. Connection edges ($CE1$ and $CE2$) and labels ($CL1$ and $CL2$) are represented as,

$$\begin{aligned} CE1 &= \{(d, s2, g3), (d, s3, g18), (d, s3, g2) \\ &\quad (d, s3, g2), (d, s4, g15)\} \\ CL1 &= \{G\_to\_E, value, E\_to\_Rel, \\ &\quad C\_to\_Rct, Rct\_to\_C\} \\ CE2 &= \{(d, s3, g14), (d, s4, g14), (d, s4, g12), \\ &\quad (d, s5, g9), (d, s8, g14), (d, s8, g13)\} \\ CL2 &= \{value, C\_to\_Rct, Rct\_to\_C, \\ &\quad G\_to\_E, C\_to\_Rct, Rct\_to\_C\} \end{aligned}$$

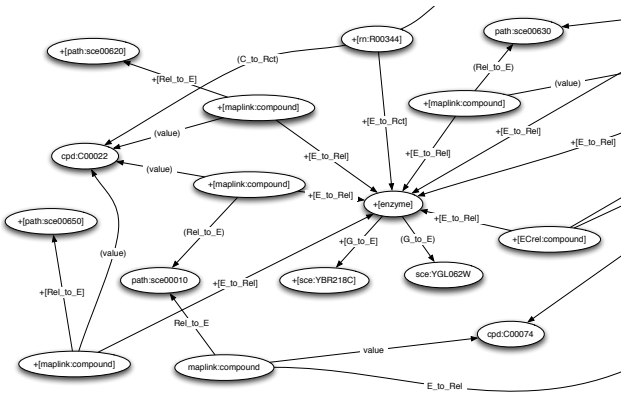

Fig. 8. A visualization of discovered subgraphs of addition rules in a dynamic graph representing the TCA cycle pathway. Labels marked by "+[]" represent the addition rules and labels marked by "()" represent the connection edges.

Vertex numbers in CE1 and CE2 represent the vertex numbers in virtual graphs, which are created at the start of the DynGRL algorithm, to specify the location to link the substructures to the parent graphs.

Figure 8 shows our visualization result of a substructure of an addition rule. The labels marked by "+[]" represent the labeled vertices and edges belonging to the substructure of the addition rule. The labels are marked by "-[]" in the case of removal rules. Connection edges between the discovered substructures and parent graphs are marked by "()". The DynGRL system helps to visualize removal or addition rules on the parent graph with the connection edges.

This result shows how the substructures in graph rewriting rules are structurally connected to the parent graphs and how the graphs change after removal or addition rules are applied. It allows us to better understand structural properties while the graphs change over time.

We perform DynGRL with a dynamic graph representing the TCA cycle and MAPK pathway for 36 time series and discover the whole set of graph rewriting rules for removals and additions. Results show the temporal relations of genes and the enzyme-enzyme relations, which are removed and added periodically with the three cycles or in order. Also, we can identify how the discovered structures connected to the parent graphs with connections edges using our visualization. These temporal patterns and graph rewriting rules help us to understand dynamic properties as well as structural properties of biological networks.

VI. Conclusion

This research defines graph rewriting rules of a dynamic graph representing structurally changing biological networks. We present the dynamic graph-based relational learning algorithm, DynGRL, to discover graph rewriting rules in a dynamic graph. The algorithm is evaluated with the dynamic graphs representing the TCA cycle metabolic pathway and MAPK pathway in combination with microarray data. The static graph represents only structural properties of biological networks, and microarray data represents only dynamic properties. Our approach can represent both properties at the same time, and discover novel patterns temporally and structurally.

We discover several interesting temporal patterns in graph rewriting rules of the metabolic pathways such that some relations between genes are shown periodically or in order. Our results are visualized to identify how the biological networks change structurally over time and what temporal patterns are discovered repeatedly. Our approach allows us to identify not only structural changes of metabolic pathways but also temporal patterns between multiple structural changes, providing us better understanding of how biological networks change over time.

The future works follow several directions. The primary challenges are to define systematic measures to assess discovered graph rewriting rules and to compare a dynamic graph generated by learned rewriting rules with a dynamic graph from real world data. We will also develop approaches to learn general patterns in the discovered rewriting rules to predict future structure of biological networks and simulate the biosystems.

References

[1] K. Nielsen, P. Sørensen, and F. H. H.-G. Busse, "Sustained oscillations in glycolysis: an experimental and theoretical study of chaotic and complex periodic behavior and of quenching of simple oscillations," *Biophysical Chemistry*, vol. 7, pp. 49–62, 1998.

[2] H. C. Causton, J. Quackenbush, and A. Brazma, *A Beginner's Guide Microarray Gene Expression Data Analysis*. Blackwell, 2003.

[3] D. J. Lockhart and E. A. Winzeler, "Genomics, gene expression and dna arrays," *Nature*, vol. 405, pp. 827– 836, 2000.

[4] J. Kukluk, C. You, L. Holder, and D. Cook, "Learning node replacement graph grammars in metabolic pathways," in *Proceedings of BIOCOMP*, 2007.

[5] C. You, L. Holder, and D. Cook, "Application of graph-based data mining to metabolic pathways," in *Proceedings of IEEE ICDM Workshop on Data Mining in Bioinformatics*, 2006.

[6] J. F. Roddick and M. Spiliopoulou, "A survey of temporal knowledge discovery paradigms and methods," *IEEE Transactions on Knowledge and Data Engineering*, vol. 14, no. 4, pp. 750–767, 2002.

[7] T. Ho, C. Nguyen, S. S. Kawasaki, S. Le, and K. Takabayashi, "Exploiting temporal relations in mining hepatitis data," *Journal of New Generation Computing*, vol. 25, pp. 247–262, 2007.

[8] M. Farach-Colton, Y. Huang, and J. L. L. Woolford, "Discovering temporal relations in molecular pathways using protein-protein interactions," in *Proceedings of RECOMBx*, 2004, pp. 150–156.

[9] H. Dörr, *Efficient Graph Rewriting and Its Implementation*. Springer, 1995.

[10] G. Rozenberg, *Handbook of Graph Grammars and Computing by Graph Transformation*. World Scientific, 1997.

[11] K. Nupponen, "The design and implementation of a graph rewrite engine for model transformations," Master's thesis, Helsinki University of Technology, Dept. of Comp. Sci. and Eng., May 2005.

[12] "KEGG website," *http://www.genome.jp/kegg/pathway*.

[13] D. Cook and L. Holder, "Substructure discovery using minimum description length and background knowledge," *Journal of Artificial Intelligence Research*, vol. 1, pp. 231–255, 1994.

[14] ——, "Graph-based data mining," *IEEE Intelligent Systems*, vol. 15, no. 2, pp. 32–41, 2000.

[15] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman & Co., 1979.

[16] B. P. Tu, A. Kudlicki, M. Rowicka, and S. McKnight, "Logic of the yeas metabolic cycle: Temporal compartmentalization of cellular processes," *Science*, vol. 310, pp. 1152–1158, 2005.

[17] D. L. Nelson and M. M. Cox, *Lehniger Principle of Biochemistry*, 4th ed. New York: Freeman, 2005.

[18] M. Qi and E. A. Elion, "Map kinase pathways," *Journal of Cell Science*, vol. 118, pp. 3569–3572, 2005.