Dynamic Graph-based Relational Learning of Temporal Patterns in Biological Networks Changing over Time

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

ABSTRACT

We propose a dynamic graph-based relational learning approach using graph-rewriting rules to analyze how biological networks change 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 to promote reproduction and sustain our lives. Most current graph-based data mining approaches overlook dynamic features of biological networks, because they are focused on only static graphs. First, we generate a dynamic graph, which is a sequence of graphs representing biological networks changing over time. Then, our approach discovers graph-rewriting rules, which show how to replace subgraphs, between two sequential graphs. These rewriting rules describe the structural difference between two graphs, and describe how the graphs in the dynamic graph change over time. Temporal relational patterns discovered in dynamic graphs representing synthetic networks and metabolic pathways show that our approach enables the discovery of dynamic patterns in biological networks.

Categories and Subject Descriptors

I.2.6 [**Artificial Intelligence**]: Learning; J.3 [**Life and Medical Science**]: Biology and genetics—*Biological Networks*

Keywords

Temporal Graph Mining, Graph Rewriting Rules, Biological Network

1. INTRODUCTION

Our bodies are well-organized biological networks, which promote reproduction and sustain our lives. Furthermore, biological networks continuously change their structures and properties, while an organism performs various biological activities, such as digestion, respiration and so on. For this reason, the analysis of dynamic biological networks is necessary to understand life at the system-level.

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.

For these reasons, we need a novel algorithm to discover structural features in accompany with temporal features in the temporal sequence of biological networks. We introduce a dynamic graph-based data mining approach using 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. After discovery of whole sets of graph rewriting rules from a dynamic graph, we discover temporal relations of graph rewriting rules. The temporal relations show what graph rewriting rule is applied before or after the other is applied. The graph rewriting rules can describe the structural difference between two graphs. The temporal relations of rewriting rules can describe how the graphs in the dynamic graph evolve over time. This approach enables us to investigate dynamic patterns in biological networks.

First, we introduce several preceding approaches related with dynamic analysis of biological networks, graph rewriting rules and temporal data mining. Then, we define the problem of our research. We represent our Dynamic Graph Relational Learning (DynGRL) algorithm. We apply our approach to synthetic biological networks and the glycolysis metabolic pathway in combination of the mathematical modeling. The results section shows our discovered graph rewriting rules

and temporal relations of rewriting rules. We also discuss biological meaning of our temporal relations and substructures.

The goal of this research is, first, to discover graph rewriting rules in a dynamic graph representing metabolic pathways changing over time. The second is to discover novel temporal relations of the graph rewriting rules to describe structural changes of graphs in a dynamic graph.

2. MOTIVATION

2.1 Dynamic Analysis of Biological Network

According to the central dogma in molecular biology, the genetic information in DNA is transcribed into RNA (transcription) and protein is synthesized from RNA (translation). Proteins play central roles in function and structure of our organisms. However, there is no biomolecule (i.e, DNA, RNA, protein, and so on) works alone. For an example, a glycolysis ($Glucose + 2NAD^+ + 2ADP + 2P_i \rightarrow 2Pyruvate + 2NADH + 2^+ + 2ATP + 2H_2O$), which is a metabolic pathway converting one molecule of glucose into two molecules of pyruvate with the production of two molecules of ATP (Adenosine TriPhosphate), includes more than 10 biochemical reactions and various enzymes [18]. Biological networks consist of various molecules and their relationships. Each molecule can have its own properties and can also influence relationships with other molecules. Currently, we categorize biological networks into three types: metabolic pathway, protein-protein interaction and gene regulatory network [12].

Two researches have approached the analysis of biological networks. One approach is graph-based data mining [13, 26]. This approach represents biological networks as graphs, where vertices represent molecules and edges represent relations between molecules and discovers frequent patterns in graphs. Many approaches of graph-based data mining discover successfully structural features of biological networks, but they overlook dynamic properties. The other approach is mathematical modeling, which is an abstract model to describe a system using mathematical formula [19]. Most of these approaches, as a type of quantitative analysis, model several kinetics of pathways and analysis the trends of amount of molecules and flux of biochemical reactions. But most of them disregard relations among multiple molecules.

There are two main aspects to consider for understanding biological networks. First, we need to focus on relations 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 aspect but also for dynamic aspect for system-level understanding of our organisms. For this reason, we need an approach to analyze graphs which structurally change over time for both aspects: structural and dynamic properties.

2.2 Graph Rewriting Rules

A graph, a set of vertices and edges between two vertices, is a distinguished way to represent a data and solve a problem in many aspects [6]. In the view of computer science, the

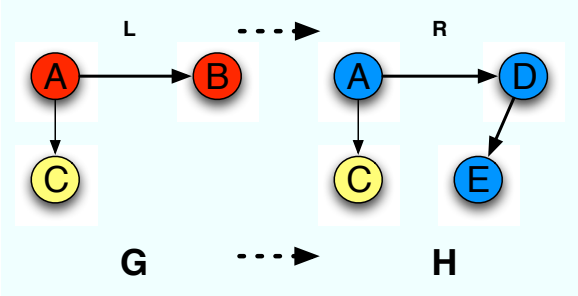

Figure 1: An example of application of graph rewriting rules, where the rule derives a graph H from a graph G by replacing a subgraph L by a subgraph R.

graph is a natural way to represent various data in practical life. Unlike the other strict data structures (i.e. linked list), graphs can define their rules to represent data using vertices and edges from the fundamental computer science to new applications like bioinformatics. In the view of mathematics, graph theory provides explicit ways to solve the various problems like the graph matching problem. In additions, graphs present amazing ways for the data visualization. Data represented as graphs are much more understandable than any other data structures. In spite of complexity issues of some graph algorithms, we have great advantage to allow users to understand data and the detail process of algorithms.

Graph rewriting is a method representing topological changes of graphs using graph rewriting rules [6, 22]. Generally, graph rewriting rules discover substructures in a graph and alter their structures. Each graph rewriting rule defines a rule between L and R, where L and R are subgraphs on two graphs G and H respectively, such that L is replacing by R, L is deleted, or R is created [20]. As shown in figure 1, L is identified first at graph G. Then L is replaced by R in graph H. Because the first task is to identify the substructure L at graph G for applying graph rewriting rules, subgraph isomorphism is the first challenge of graph rewriting rules. Because graph rewriting rules represent dynamic properties of graphs, they are used for the operational specification of systems, such as planning task, software development environments, transaction modeling in database, and so on [6, 23].

There are also several algorithms to discover the node or edge replacement graph grammar using the minimum description length principle [11, 14]. However, their scope is limited to static graphs.

A few approaches have applied graph rewriting rules to analyze biological networks [2, 7]. Andrei et al. [2] propose a graph rewriting strategy for modeling biochemical networks. They encode molecules as terms, reaction patterns as rewriting rules and the transformations on molecules as rewriting relations. Using these encoding schemes, they model the control mechanism in biochemical networks. Biochemical networks are described as the encoded equation in their schemes. They are focused on encoding the reactions, but overlook structural properties of networks. A graph trans-

formation method [7] is used for tracing molecules during sequential reactions of a metabolic pathway. They represent each molecule in the pathway as a hypergraph. Then they apply their graph transformation for each chemical reaction between molecules. During the transformation, they map the previous states of atoms to the next states in each molecule to trace the history of atoms or molecules. This way is focused on tracing the history atoms and molecules rather than structural changes.

2.3 Temporal Data Mining

Temporal data mining is a type of knowledge discovery approach to mine temporal patterns rather than static such as discovery of temporal relations or cause-effect association [21]. Temporal data mining mainly discovers patterns in sequential data, which is ordered with respect with some index like time stamps [15]. In temporal data mining, we have two major tasks: prediction of the values in the future and discovery of temporal patterns [21]. More importantly, temporal data mining is able to discover relational aspects of data. In other words, we can understand how or why the object changes rather than merely what object.

In this research, we are focused on discovery of temporal patterns and their visualization. Mörchen's research [17] shows several unsupervised learning techniques for discovery of temporal patterns using symbolic temporal data. Laxman and et al. [16] introduce an algorithm find frequent episodes in event streams. This approach counts the number of frequency based on non-overlapped occurrences of the episodes. Allen and et al. [1] formalized temporal logic for time intervals using 13 interval relations. This approach allows us to present temporal relations of sequential data.

Several methods have addressed data mining in a dynamic graph. GraphScope [24] is a technique to discover communities and detect changes in graphs changing over time. They use matrix and encoding schemes to represent a dynamic graph. However, they limit their scope to the bipartite graphs. The other approach [3] proposes several detection measures of abnormal changes in the sequence of graphs. Among these approaches, there are few approaches to describe how the graphs change over time, where they detect just what changes.

There are several researches to apply temporal data mining in biological data. Ho and at al. [10] propose an approach to detect temporal patterns and relations between medical events of Hepatitis data. They represent medical information of patients as sequential events. They classify temporal patterns and relations of medical testing results in the sequential events using Naive Bayes classifier. A research by Farach-Colton and et al. [8] introduce an approach of mining temporal relations in protein-protein interactions. They try to model the assembly pathways of Ribosome using protein-protein interactions, which are involved with the temporal order. This approach determine the order of molecular connections using the distance measure of each interaction between two proteins.

Our focus in this paper is temporal data mining using graph-based relational learning. Graph-based relational learning is not conductive to a quantitative analysis of temporal data,

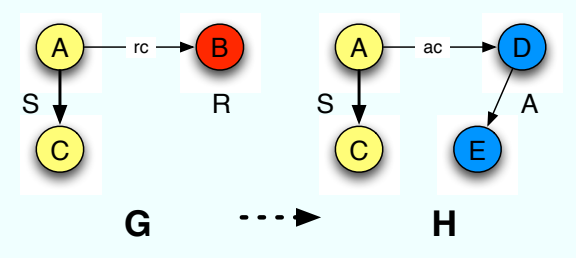

Figure 2: An example of application of graph rewriting rules, which shows an removal rule {R, rc} from a graph G and an addition rule {A,ac} to a graph H. The removal and addition substructures are connected to G and H by edges rc and ac. S represents the common subgraph between G and H.

because a graph has a limitation of representing numerical values. If we represent temporal data using temporal relations between actions and events, a graph is a great alternative to represent temporal relations because actions and events can have rich relational structures during their processing time. These relationships, such as overlapping and occurring simultaneously, among actions, events and their effects are too complex to represent without graphs. For this reason, various temporal data are eligible to be represented as graphs to describe their temporal relations.

3. PROBLEM DEFINITION

Our main goal of this research is to discover graph rewriting rules in a series of graphs changing their structures over time. Each graph rewriting rule represents topological changes between two sequential graphs. Here, we define our problem with graph rewriting rules.

As described in the previous section, traditional approaches of graph rewriting rules determine which subgraphs will be replaced by the other subgraphs. In our approach, we are focused on representing changing structures between two graphs rather than just what subgraphs change. First, we discover maximum common subgraphs between two sequential graphs. Then, we derive removal substructures from the left-handed graph and addition substructures from the right-handed graph. Figure 2 shows an instance of this process. A maximum common subgraph (marked as yellow) is discovered between two graphs, G and H. Then the remains in G and H become removal (red marked) and addition (blue marked) substructures respectively. These substructures with connection edges rc and ac are elements of graph rewriting rules: removal and addition rules respectively. For this approach, we define several preliminary terms.

First, we define a dynamic graph DG including n time series of graphs like the following.

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

Each graph G_i is a graph in time i for $1 \leq i \leq n$. G_i is transformed to G_{i+1} by application of graph rewriting rules, which will be described in this section.

Then, we define a set of removal substructures RG and a set

of additional substructures AG like the following.

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

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.

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 dose not connect to the parent graph.

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

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

A set of graph rewriting rules $GR_{i,i+1}$ between two sequential graphs G_i and G_{i+1} are defined as the combination of RG, AG, CE and CL like the following.

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

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 additions substructure in AG_{i+1} respectively. CE_k and CL_k represent connection edges between substructures ($k = m$ or n) like the following.

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

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 link 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 vertex, and the end vertex from the parent graph starts with "g" followed by the index of 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) is the number of connections from the substructures and the parent graph. If a substructure is disconnected from the parent graph, both sets of CE and CL are $\emptyset$. We will describe more detail with some examples in the result section.

4. DISCOVERY OF GRAPH REWRITING RULES

The first goal of our research is to discover graph rewriting rules in a dynamic graph representing biological networks changing over time. This section describes our algorithm to discover graph rewriting rules in a dynamic graph.

Algorithm 1 Discovery of Graph Rewriting Rules in Dynamic Graph

Require: $DynamicGraph = \{G_1, G_2, \cdots, G_n\}$,$Limit$
1. Create $VirtualGraphList = \{VG_1, VG_2, \cdots, VG_n\}$
2. $ListOfRewriteRules = \{\}$
3. **for** $i = 1$ to $n - 1$ **do**
4. $RemRuleSet$ =AddRuleSet = $ComSubSet = \{\}$
5. $Graphs = \{G_i, G_{i+1}\}$
6. $Limit = UV + 4(UE - 1)$
7. **while** No more compression **do**
8. $BestSub$ = DiscoverSub($Limit, Graphs$)
9. **if** $BestSub$ is common in both G_i and G_{i+1} **then**
10. Add $BestSub$ into $ComSubSet$
11. **end if**
12. Compress $Graphs$ by BestSub
13. Mark $BestSub$ on VG_i and VG_{i+1}
14. **end while**
15. Get $removalSubs$, $conEdges$ from VG_i
16. Add $removalSubs$ into $RemSubSet$ and $conEdges$ into $RemConEdgeSet$
17. Get $additionSubs$, $connectionEdges$ from VG_{i+1}
18. Add $additionSubs$ into $AddSubSet$ and $conEdges$ into $AddConEdgeSet$
19. Create $rewriteRules$ from $RemSubSet$, $AddSubSet$, $RemConEdgeSet$, $AddConEdgeSet$
20. Add $rewriteRules$ into $ListOfRewriteRules$
21. **end for**
22. **return** $ListOfRewriteRules$

4.1 Algorithms

Our algorithm starts with a dynamic graph including n time series of graphs as shown in algorithm 1. First, our algorithm creates a list of n virtual graphs 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 specifies $limit$ based on label information of G_i and G_{i+1} (line 6). UV and UE denote the number of unique vertex labels and edges in G_i and G_{i+1}. $limit$ specifies the number of substructures to consider in each iteration at SUBDUE process.

An inner loop (line 7 to 14) represents procedures to discover common substructures between two sequential graphs. We uses SUBDUE graph-based relational learning approach to discover substructures [4, 5]. SUBDUE evaluates substructures using Minimum Description Length (MDL) principle. More detail evaluation approach is described in [4]. After discovery of the best substructure, the algorithm checks whether the substructure is the subgraphs of the both graphs G_i and G_{i+1}. In the affirmative case, the best substructure is added into $ComSubSet$ and two target graphs are compressed by replacing the substructure with a vertex. If the best substructure does not belong to one of 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 no more is compressed.

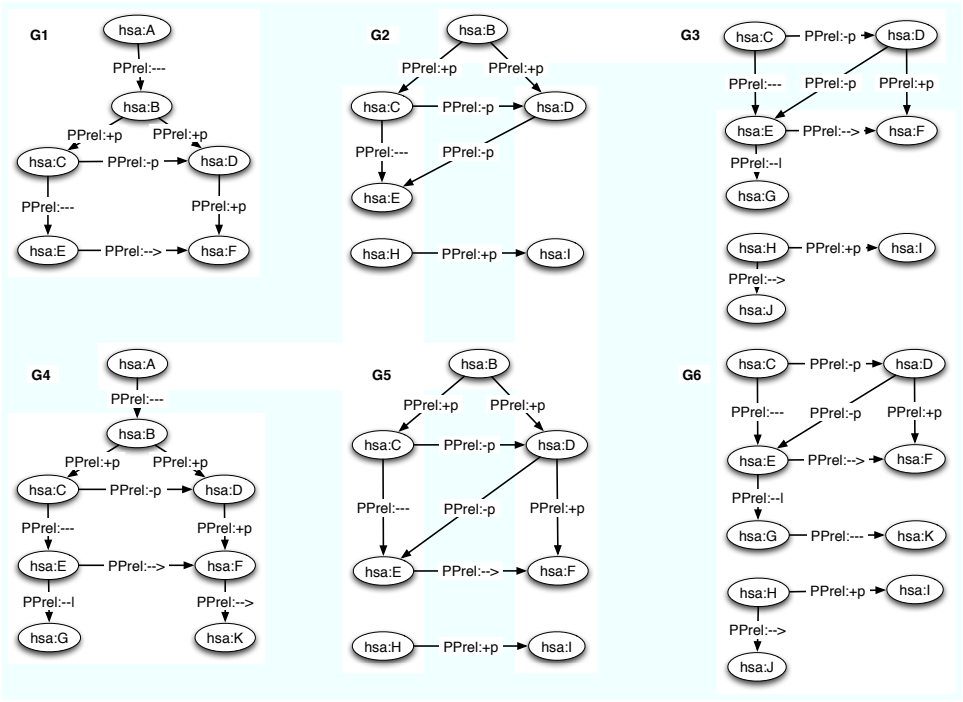

Figure 3: An example of dynamic graph including 6 time series of graphs representing artificial biological networks.

Using the complete list of common substructures, $ComSubSet$, the algorithm acquires removal substructures and addition substructures (line 15 and 17). First, the algorithm marks vertices and edges used in common substructures and find each disconnected 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 for visited or to be visited. The marked 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 connections edges, $conEdges$, as described in the previous section. These edges are added into $RemConEdgeSet$ and $AddConEdgeSet$, when 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).

4.2 Complexity Issue

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 [9]. 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 change over time is not so big. Therefore, discovery of graph rewriting rules is still feasible. However, we have still challenges to analyze a huge size of graphs.

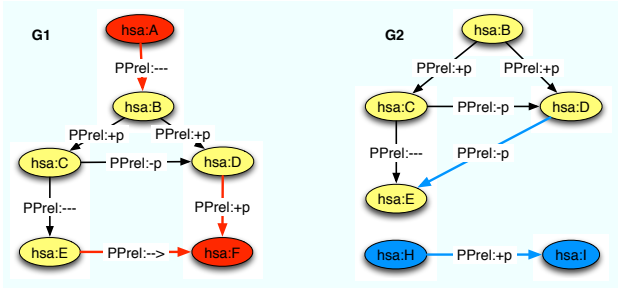

Figure 4: An instance of graph rewriting rules between graph G1 and G2 in the synthetic biological networks

5. EXPERIMENTS

We prepare two types of dynamic graphs to evaluate our algorithm. The first one is a synthetic dynamic graph including 6 time series of graphs. The second one is a dynamic graph representing the glycolysis metabolic pathway in combination with a mathematical modeling result.

5.1 Synthetic Biological Network

Figure 3 shows a dynamic graph including 6 time series of graphs, which represent synthetic biological networks. These networks represent artificial protein-protein interactions, where vertices represent proteins and edges represent interactions between two proteins. Figure 4 shows an instance of graph rewriting rules between G_1 and G_2. The yellow-colored subgraphs in both graphs represent common substructures. The red elements represent removal substructures (from G_1) with connection edges and the blue elements

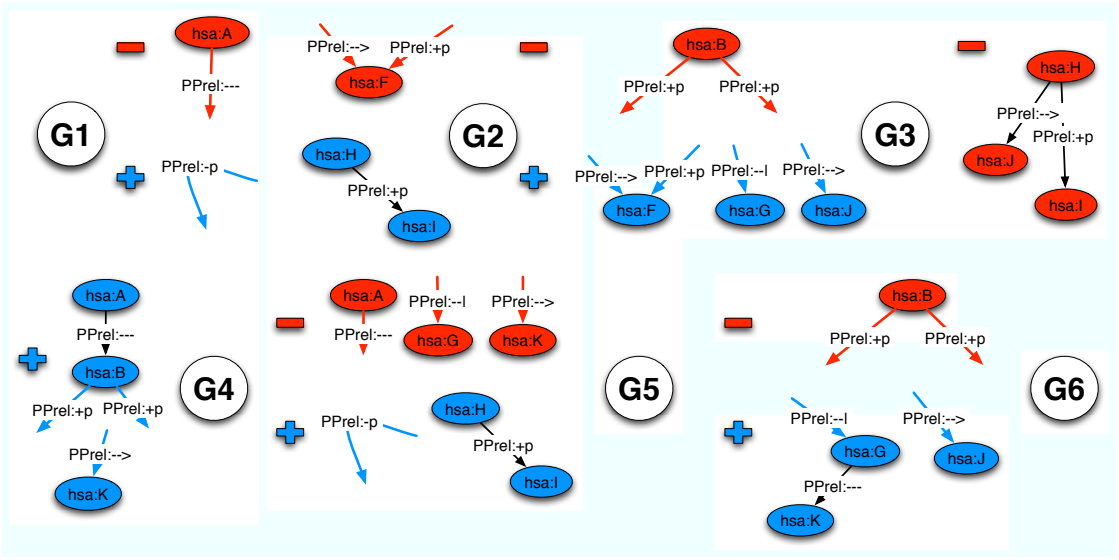

Figure 5: Temporal patterns of dynamic graph. Orange subgraphs represent removal rules and green subgraphs represent addition rules. Orange edges and green edges represent connection edges for removal and addition rules respectively.

represent addition substructures (to G_2) with connection edges.

$GR_{1,2}$ 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 in the previous section. It has four 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 A. $rSub_1$ was connected by an edge $(d, s1, g2)$, which is labeled by $PPrel : ---$. This edge is a directed edge (noticed as 'd'). One end of this edge is $s1$, which denotes a vertex number 1 in $rSub_1$ (s denotes the substructure.). The other end is $g2$, which denotes a vertex number 2 in G_1 (g denotes the original graph.). a_1 and a_2 represent addition rules similarly. But these two cases look somewhat different. a_1 has $\emptyset$ as a addition substructure, because a_1 is a rule representing a blue edge $PPrel : -p$ of G_2 without any addition substructure as shown in figure 4. a_2 has also $\emptyset$s for edges and edge labels, because $aSub_1$ represents a disconnected graph including vertices H and I in G_2.

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

Figure 5 shows all sets of graph rewriting rules in five sequential intervals. Red elements represent removal rules from G_i for $i = 1$ to 5, and blue elements represent addition rules to G_j for $j = 2$ to 6. G_1 and G_6 do not have removal rules and addition rules respectively, because they are starting and end graphs. In this result, we can easily discover several temporal relations of graph rewriting rules. A subgraph including a vertex A is always removed right before a subgraph including a vertex B is removed. These patterns are observed in from G_1 to G_3 and from G_4 to G_6. Vertices G and J are added right after a subgraph including vertices H and I is added. These patterns are also discovered in from G_1 to G_3 and from G_4 to G_6. These temporal relations can describe how the graphs change over time as well as what structures change.

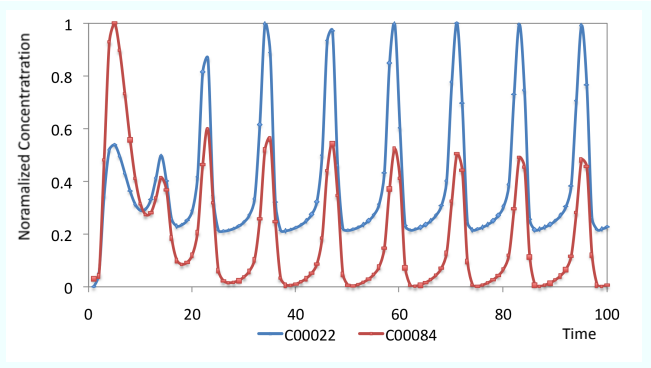

Figure 6: The oscillation curves of changing concentrations of Pyruvate (C00022) and Acetaldehyde (C00084).

5.2 Glycolysis Pathway

The glycolysis pathway, which is the first fully investigated metabolic pathway, a series of enzyme-catalyzed reactions of degrading a molecule of glucose (six carbons) to yield two molecules of pyruvate (three carbons) [18]. In the process of glycolysis, some of the free energy are released from glucose and converted to the form of ATP and NADH. Glycolysis is the most important pathway in aerobic respiration, which is a process to generate energy in a cell.

As described in the previous section, a mathematical mod-

eling approach explores only numerical values, such as the concentration of molecules and the flux of reactions. We here propose to combine the result of mathematical modeling and graphs for structural analysis of changes over time. We use a result of the simulation of glycolysis pathway of the yeast (*Saccharomyces cerevisiae*) [19]. This result contains the trends of concentrations of 14 molecules. We normalize these concentrations from 0 to 1, because we are focused on trends of the changes and the concentration of different molecules are various. Figure 6 shows the oscillated curves of the normalized concentration of two molecules: Pyruvate and Acetaldehyde. Because the simulation is performed for 100 seconds, we have 101 time series data from the initial time to the final time.

We generate a static graph representing the glycolysis pathway from KEGG PATHWAY database [25] where vertices represent reactions, compounds and enzymes, and edges represent relations between vertices. Usually, a reaction catalyzed by one or more enzymes contains one or more compounds as substrates and one or more compounds as products. We use a threshold t to activate compounds. At each time, we assume a compound, which has more than t amount, is shown in the graph. Consequently, the reactions and enzymes are shown in the graph, when all of connected substrates and products are activated at the time. In other words, every related compound should be activated to place a reaction in a graph. We try 0.1 and 0.3 as our thresholds.

We perform DynGRL with a dynamic graph of 101 graphs representing glycolysis simulation for 101 seconds. Then, we have 100 sets of graph rewriting rules during 100 time intervals for each threshold: 0.1 and 0.3. First, we discuss temporal relations of graph rewriting rules. Then, we represent discovered substructures, which belong to graph rewriting rules.

5.2.1 Temporal Relations of Graph Rewriting rules

As described in the previous section, the goal of this research discovers temporal relations of graph rewriting rules to describe structural changes of metabolic pathways over time. Because the modeling result represents the oscillation of glycolysis, we observe several temporal relations of graph rewriting rules.

In both experiments (threshold 0.1 and 0.3), we discover several temporal relations. Using 0.1 as the threshold, temporal relations among three chemical reactants such as C00008 (ADP), C00084 (Acetaldehyde) and C00003 (NAD^+) are discovered as shown in figure 7 (a). The points above the time interval axis represent the time when each compound is removed from the graph representing the glycolysis pathway. The points below the axis represent the time when each compound is added to the pathway graph. For example, the first red diamond point closed to 0 represent the addition of a substructure including the ADP molecule at time 2. These time points are ordered by the time of removal and addition. Acetaldehyde is added before ADP is added except the first case, and then NAD^+ is added if applicable . In case of removal, they are removed following the order of ADP, Acetaldehyde and NAD^+.

Similarly, figure 7 (b) shows other temporal relations between Acetaldehyde and Pyruvate (C00022) at the experiment of threshold 0.3. We observe Acetaldehyde is added after Pyruvate is added to the pathway graph. In contrast, Pyruvate is removed after Acetaldehyde is removed from the pathway graph. We can compare these temporal relations with the mathematical modeling results shown in figure 6. The oscillation curves represent that Pyruvate is increased slightly earlier and decreased slightly later than Acetaldehyde. For this reason, the temporal relations represent the substructures including Pyruvate are added earlier and removed later. We can also discuss biological meaning of these temporal relations.

Here we have two reactions: R00224 and R00754. R00224 is a reversible reaction between Pyruvate and Acetaldehyde, and R00754 is a reversible reaction between Acetaldehyde and Ethanol. We here denote the former as a substrate and the latter as a product. The mathematical modeling [19] of our research assumes the concentration of Ethanol, which is the final product of the pathway, is increased continuously. Therefore, the last second product, Acetaldehyde, is also consumed continuously to produce Ethanol. However, the production of Acetaldehyde is dependent on the amount of Pyruvate. According to Le Châtelier's principle, we can predict the decreasing (increasing) amount of the product using the increasing (decreasing) amount of the substrate. Because Pyruvate is a substrate of the reaction to produce Acetaldehyde, Pyruvate increases earlier and decreases later than Acetaldehyde. In other words, the increment of Pyruvate causes the increment of Acetaldehyde, and the decrement of Acetaldehyde causes the decrement of Pyruvate. For the production of Ethanol continuously, Acetaldehyde cannot reach its equilibrium state (maximal points in the oscillation curves) of the reaction before Pyruvate reach its equilibrium state (at most same time). Otherwise, Ethanol is decreased.

In this experiment, we show that DynGRL discovers graph rewriting rules from a dynamic graph representing the glycolysis pathway changing over time. These graph rewriting rules represent temporal relations that describe how the structure of the glycolysis pathway changes over time by showing which elements changes earlier than the other. These temporal relations and graph rewriting rules help us to understand dynamic properties of the glycolysis pathway.

5.2.2 Discovered Substructures

The other goal of this research is to show structural changes of metabolic pathways as well as temporal relations. Because an advantage of the graph representation is visualization, we can understand metabolic pathways better using structural analysis with temporal analysis. This section represents an instance of discovered substructures with graph rewriting rules.

Figure 8 shows a substructure of removal rules discovered at the experiment of threshold 0.3. The instances of this substructure are removed from the pathway at time 24, 49, 60, 72, 84, and 96. As shown in figure 7 (b), there are six time points (the third point and the last five points above the axis) when Acetaldehyde and Pyruvate are removed at the same time. The substructure in figure 8 represents the removal substructures at these six time points.

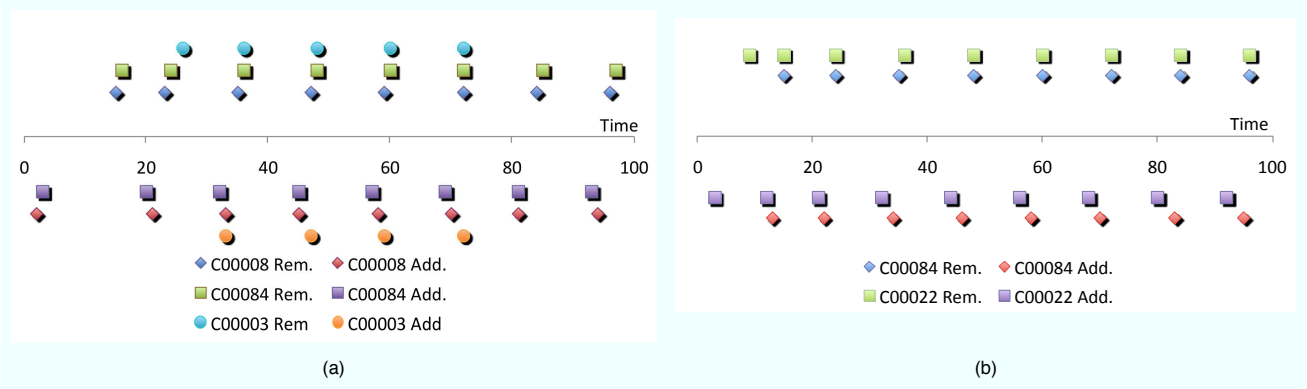

Figure 7: A visualization of time points when the substructure including each compound is removed from or added to graphs representing the glycolysis pathway at the experiment of threshold 0.1 (a) and 0.3 (b).

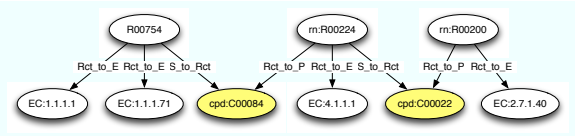

Figure 8: An instance of substructure included in removal rules.

In this substructure, Acetaldehyde (C00084) works as a substrate of a reaction R00754 and as a product of a reaction R00224. Pyruvate (C00022) works as a substrate of the reaction R00224 and as a product of the reaction R00200. From the description in the previous section, we easily assume R00224 is the reaction of converting Pyruvate to Acetaldehyde like the following equation.

$$R00224 : C00022 \leftrightarrow C00084 + C00011(CO_2)$$

Other two reactions represent the following chemical reactions.

$$R00754 : C00469 + C00003 \leftrightarrow C00084 + C00004 + C00080$$

C00469 denotes Ethanol, C00003 denotes NAD^+, C00004 denotes $NADH$ and C00080 denotes H^+.

$$R00200 : C00002 + C00022 \leftrightarrow C00008 + C00074$$

C00002 denotes ATP, C00008 denotes ADP and C00074 denotes Phosphoenolpyruvate.

These three reactions and the four enzymes EC: 1.1.1.1, EC: 1.1.1.71, EC: 4.1.1.1 and EC: 2.7.1.40 are removed with Acetaldehyde and Pyruvate, because after removing both chemical compounds these reactions and enzymes are useless. Notice that we assume a reaction and enzyme are shown when all related reactants (compounds) are activated. In each time, before removal, the reactions and enzymes are shown because all related reactants are activated including C00022 and C00084. When both chemical compounds are removed, these reactions and enzymes are also removed because there is nothing to react. In case of CO_2 and H^+, we exclude the node of CO_2 and H^+ in our graphs because the mathematical model excludes these two compounds.

$GR_{24,25}$ including the substructure index is represented as,

$$GR_{24,25} = \{(r_1, rSub_1, CE, CL)\}$$

where $rSub_1$ represents the substructure in figure 8 and CE and CL represent a set of connection edges and connection edge labels like the following,

$$\begin{aligned} CE &= \{(d, s2, g5), (d, s2, g9), (d, s2, g11), \\ &\quad (d, s7, g8), (d, s7, g10), (d, s7, g12)\} \\ CL &= \{(S\_to\_Rct, S\_to\_Rct, Rct\_to\_P, \\ &\quad Rct\_to\_P, S\_to\_Rct, S\_to\_Rct\} \end{aligned}$$

As described in previous section,"g" denotes the original graph and "s" denotes the substructure. The six connection edges are connected six vertices in the graph at time 24 such as C00074, C00002, C00008, C000469, C00004 and C00003, which work as substrates or products in the above three reactions. The other ends of connection edges are connected to the vertices of the denoted indices in the removal substructure. Figure 9 shows a graph at time 24, where the red marked subgraph represents a substructures shown in figure 8 and the blue marked edges represent the connection edges as the elements in CE and CL

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 understand better structural properties while the graphs change over time.

6. CONCLUSION

In this research, we formalize graph rewriting rules of graphs representing structurally changing biological networks. We represent an algorithm, DynGRL, to discover graph rewriting rules in a dynamic graph. The algorithm is evaluated with the synthetic dynamic graph and the dynamic graph representing the glycolysis metabolic pathway in combination with mathematical modeling results. We also discover several interesting temporal relations of graph rewriting rules of the metabolic pathways. Our results are visualized to identify how the metabolic pathways change structurally over time and what temporal patterns are discovered repeatedly. Our approaches allow us to identify not only structural changes of metabolic pathways but also temporal relations

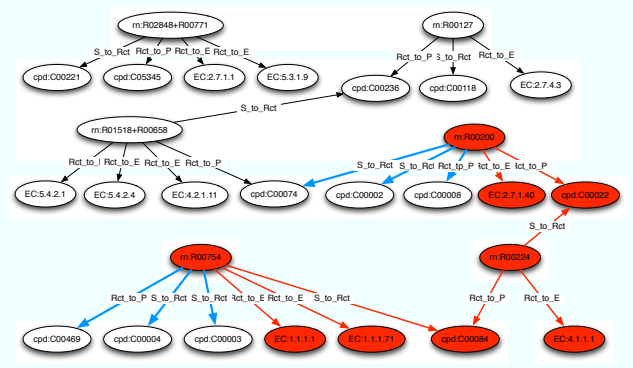

Figure 9: An example of graph at time 24. Red marked subgraphs represents a removal substructure as shown in figure 8 and blue edges represent connection edges.

between multiple structural changes, providing us better understanding of how biological networks changes over time.

Our future works include the development of evaluation methods and application to other networks such as gene regulatory networks in combination with microarray data. To evaluate discovered graph rewriting rules is not straightforward but necessary. The primary issues are to define systematic measures to assess discovered graph rewriting rules and to compare our regenerated graphs by rewriting rules with real world data. The graph representation of the non-structural data, like microarray, is also required.

7. REFERENCES

[1] J. F. Allen and G. Ferguson. Actions and events in interval temporal logic. *Journal of Logic and Computation*, 4:531–579, 1994.

[2] O. Andrei and H. Kirchner. Graph rewriting and strategies for modeling biochemical networks. In *Proceedings of the International Symposium on Symbolic and Numeric Algorithms for Scientific Computing*, pages 407–414, 2007.

[3] H. Bunke, M. Kraetzl, P. Shoubridge, and W. Wallis. Detection of abnormal change in time series of graphs. *Journal of Interconnection Networks*, 3, Nos 1+2:85–101, 2002.

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

[5] D. Cook and L. Holder. Graph-based data mining. *IEEE Intelligent Systems*, 15(2):32–41, 2000.

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

[7] K. Ehrig, R. Heckel, and G. Lajios. Molecular analysis of metabolic pathway with graph transformation. In *ICGT*, pages 107–121, 2006.

[8] M. Farach-Colton, Y. Huang, and J. L. L. Woolford. Discovering temporal relations in molecular pathways using protein-protein interactions. In *Proceedings of the international conference on Resaerch in computational molecular biology*, pages 150–156, 2004.

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

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

[11] I. Jonyer, L. Holder, and D. Cook. Mdl-based context-free graph grammar induction. In *Proceedings of FLAIRS-2003.*, 2003.

[12] H. Kitano. Systems biology: A brief overview. *Science*, 295:1662–1664, 2002.

[13] J. Kukluk, C. You, L. Holder, and D. Cook. Learning node replacement graph grammars in metabolic pathways. In *Proceedings of International Conference on Bioinformatics and Computational Biology*, 2007.

[14] J. P. Kukluk, L. B. Holder, and D. J. Cook. Inference of node replacement recursive graph grammars. In *Proceedings of the Sixth SIAM International Conference on Data Mining*, 2006.

[15] S. Laxman and P. S. Sastry. A survey of temporal data mining. *Sadhana*, 31(2):173–198, 2006.

[16] S. Laxman, P. S. Sastry, and K. P. Unnikrishnan. A fast algorithm for finding frequent episodes in event streams. In *Proceedings of the 13th ACM SIGKDD*, pages 410–419, 2007.

[17] F. Mörchen. Unsupervised pattern mining from symbolic temporal data. *SIGKDD Explorations*, 9:41–55, 2007.

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

[19] 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*, 7:49–62, 1998.

[20] K. Nupponen. The design and implementation of a graph rewrite engine for model transformations. Master's thesis, HELSINKI UNIVERSITY OF TECHNOLOGY, Department of Computer Science and Enggineering, May 2005.

[21] J. F. Roddick and M. Spiliopoulou. A survey of temporal knowledge discovery paradigms and methods. *IEEE Transactions on Knowledge and Data Engineering*, 14(4):750–767, 2002.

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

[23] N. Sarkar and M. S. Sarkar. A graph-rewriting approach to high-level task planning - an introduction. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 661–666, October 1999.

[24] J. Sun, C. Faloutsos, S. Papadimitriou, and P. S. Yu. Graphscope: parameter-free mining of large time-evolving graphs. In *Proceedings of the 13th ACM SIGKDD*, pages 687–696, 2007.

[25] KEGG website. *http://www.genome.jp/kegg/pathway*.

[26] 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.