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

Chang Hun You, Lawrence B. Holder and Diane J. Cook
School of Electrical Engineering & Computer Science, Washington State University, Pullman, WA, USA
{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 patterns discovered in dynamic graphs representing metabolic pathways show that our approach enables the discovery of dynamic patterns in 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. We assume the structures of biological networks change over time as they interact with specific conditions, for instance, a disease.

The temporal patterns in the structural changes of biological networks can be significant information about a disease and help researchers develop new drugs. During the development period, the temporal patterns in the structural changes of biological networks after taking the medicine are also used for the development and evaluation of the new drug. Lactose intolerance is the inability to digest lactose because of a lack of the lactase enzyme, breaking down lactose into galactose and glucose [2]. Two major treatments are to minimize the intake of lactose products and take the lactase supplement. Our approach can help us discover the temporal patterns in the structural changes of the galactose metabolism pathway after these treatments, and investigate other treatments (i.e., improving the production of the lactase enzyme in the pathway).

We propose a novel approach to analyze structural features along with temporal features in a time series of biological networks to enhance our systems-level understanding of bio-organisms. Our dynamic graph-based data mining 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. After discovery of whole sets of graph rewriting rules from a dynamic graph, we discover temporal patterns in graph rewriting rules. The temporal patterns 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 patterns in rewriting rules can describe how the graphs in the dynamic graph changing over time. This approach enables us to investigate dynamic patterns in biological networks.

This paper, first, introduce, several preceding approaches related to dynamic analysis of biological networks and temporal data mining. Then, we define the problem of our research. We present our Dynamic Graph Relational Learning (DynGRL) algorithm. Our approach is applied to the glycolysis metabolic pathway in combination with the mathematical modeling. The results section shows our discovered graph rewriting rules and temporal patterns of rewriting rules in two aspects: temporal and structural aspects.

2 Related Works

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). These biomolecules (DNA, RNA and proteins) play central roles in the aspects of the function and structure of organisms. However, there are few molecules that can work 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 [13]. Biological networks including metabolic pathways, protein-protein interactions and gene regulatory networks, consist of various molecules and their relationships [10]. 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.

Two approaches have been developed for the analysis of biological networks. One approach is graph-based data mining [11, 18]. 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 structural features of biological networks, but they overlook temporal properties. The other approach is mathematical modeling, which is an abstract model to describe a system using mathematical formulae [14]. Most of these approaches, as a type of quantitative analysis, model the kinetics of pathways and analyzes the trends in the amounts of molecules and the flux of biochemical reactions. But most of them disregard relations among multiple molecules.

Temporal data mining attempts to learn temporal patterns in sequential data, which is ordered with respect to some index like time stamps, rather than static data [16]. Temporal data mining is focused 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 approaches to apply temporal data mining in biological data. Ho et al. [8] propose an approach to detect temporal patterns and relations between medical events of Hepatitis data. They represent medical information of patients as sequential events and classify temporal patterns and relations of medical testing results in the sequential events using the Naive Bayes classifier. Farach-Colton et al. [6] introduce an approach of mining temporal relations in protein-protein interactions. They model the assembly pathways of Ribosome using protein-protein interactions. This approach determines the order of molecular connections using the distance measure of each interaction between two proteins.

Temporal data mining approaches discover temporal patterns in data, but they disregard relational aspects among entities. For example, they can identify temporal patterns in the appearance of genes such that a gene, YBR218C, appears before another gene, YGL062W, but cannot identify how these two genes interact with each other.

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.

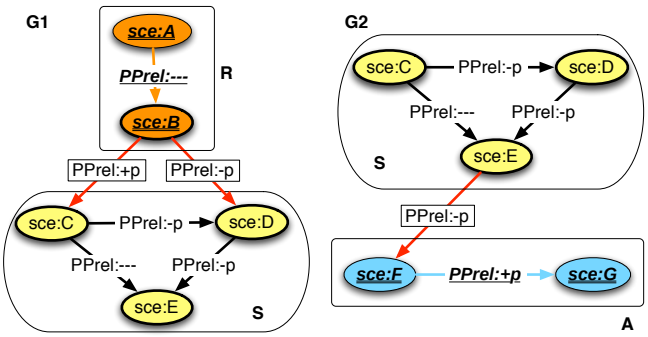

Figure 1. An example of graph rewriting rules between two graphs G_1 and G_2. S represents the maximal common subgraph between the two graphs. R and A represent the removal and addition substructures. The red edges with labels marked by the boxes represent the connection edges.

3 Problem definition

This paper focuses on temporal and structural analysis of biological networks. Our dynamic graph-based relational learning approach discovers 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 graph rewriting rules for our approach.

Graph rewriting is a method to represent topological changes of graphs using graph rewriting rules [5, 17]. 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 [15]. There are also several algorithms to discover the node or edge replacement graph grammar using the minimum description length principle [9, 12]. However, their scope is limited to static graphs.

Traditional approaches to the identification of graph rewriting rules determine which subgraphs will be replaced by other subgraphs. Our approach is focused on representing changing structures between two graphs rather than just what subgraphs change. 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 remaining structure in G_1 and G_2 becomes removal (denoted by R) and addition (denoted by A) substructures respectively. These substructures with connection edges (red edges with boxed labels) are elements of graph rewriting rules: removal and addition rules. For this approach, we define several preliminary terms.

A 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 $x, y \in V$. Here, we define a dynamic graph DG as a sequence of n graphs as $DG = \{G_1, G_2, \cdots, G_n\}$, where 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 graph G_i, AG_{i+1} denotes a set of addition substructures in the next 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 denoted by

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

A set of graph rewriting rules $GR_{i,i+1}$ between two sequential graphs G_i and G_{i+1} is defined 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 G_i (G_{i+1}). CE_k and CL_k represent connections between substructures and the original 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 the starting and ending vertices of the edge. Because the connection edge links the substructure to the original graph, one end of this edge is from the substructure and the other is from the original graph. The end vertex from the substructure starts with "s" followed by the index of the vertex, and the end vertex from the original 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 original 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 original graph. If a substructure is not connected to the original graph, both sets of CE and CL are empty.

Using the definitions of graph rewriting rules, we describe more detail about the example in figure 1. As described previously, the graph rewriting rule, $GR_{1,2}$, includes one removal rule and one addition rule. The removal rule includes one substructure (denoted by R) and two connection edges (red edges with boxed labels). The addition rule includes one substructure (denoted by A) and one connection edge. G_1 is transformed to G_2 by application of $GR_{1,2}$ ($G_2 = G_1 \bigoplus GR_{1,2}$). $GR_{1,2}$ is described as follows,

$$\begin{aligned} GR_{1,2} \quad = \quad & \{(r_1, rSub_1, \{(d, s2, g3), (d, s2, g4)\}, \\ & \{PPrel : +p, PPrel : -p\}), \\ & (a_1, aSub_2, \{(d, g3, s1)\}, \{PPrel : -p\})\}, \end{aligned}$$

where r_1 represents an index into the set of removal rules and a_1 represents an index into the set of addition rules. A removal rule r_1 includes a removal substructure $rSub_1$ denoted by R in figure 1. $rSub_1$ was connected to the original graph G_1 by two edges $(d, s2, g3)$ and $(d, s2, g4)$, which are labeled by $PPrel : +p$ and $PPrel : -p$. These connection edges are directed edges (indicated by 'd'). These two edges are connected from the substructure (denoted by 's') to the original graph (denoted by 'g'), where each number denotes a vertex number in the substructure or the original graph. For example, $(d, s2, g3)$ denotes a connection from a vertex number 2 in the substructure to a vertex number 3 in the original graph. In a similar way, an addition rule a_1 includes an addition substructure $aSub_1$ (denoted by A in figure 1), which is connected by one connection edge $(d, g3, s1)$ labeled by $PPrel : -p$.

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 what structures change.

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.

This section describes our graph rewriting rule discovery system, DynGRL, that discovers graph rewriting rules in a dynamic graph. 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 application 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 connection 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 $Limit$ specifies the number of substructures to consider when searching for a common substructure (line 6). 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.

The inner loop (lines 7 to 14) represents the procedure to discover common substructures between two sequential graphs. We use the SUBDUE graph-based relational learning approach to discover substructures [3, 4]. 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 [3]. 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 this section. 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, $addSubs$, (lines 15 and 17). First, the algorithm identifies vertices and edges not part of common substructures and finds each disconnected substructure 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 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 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 lines 16 and 18). Using these rewriting substructures and connection edges, rewriting rules (RR) are created and stored into RRL (in lines 19 to 20).

The main challenge of our algorithm is to discover maximum common subgraphs between two sequential graphs, because this problem is known to be NP-hard [7]. To avoid this problem, first we use the $Limit$ to restrict the number of substructures to consider in each iteration. The $Limit$ is computed using the number of unique labels of vertices and edges in graphs. Second, our algorithm does not try to discover the whole common substructures at once. In each step, the algorithm discovers a portion of common, connected substructure and iterates the discovery process until discovering the whole maximum common subgraphs. Usually, the size of graphs representing biological networks is not too large. Therefore, discovery of graph rewriting rules is still feasible. However, we still have challenges to analyze very large graphs.

Algorithm 1: DynGRL 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 $ComSet$
11 **end**
12 Compress $Graphs$ by $BestSub$
13 Mark $BestSub$ on VG_i and VG_{i+1}
14 **end**
15 Get $remSubs$ and CE from VG_i
16 Add $remSubs$ into $RemSubSet$ and CE into $RemCESet$
17 Get $addSubs$ and 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**

5 Dynamic Graph Generation and Experiment

We evaluate our algorithm using a dynamic graph representing the glycolysis metabolic pathway in combination with a mathematical modeling result. The glycolysis pathway is a series of enzyme-catalyzed reactions of degrading a molecule of glucose (6 carbons) to yield two molecules of pyruvate (3 carbons) [13]. In the process of glycolysis, some of the free energy are produced as the forms 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 modeling approach explores only numerical values, such as the concentration of molecules and the flux of reactions. We propose to combine the result of mathematical modeling and graphs for structural and temporal analysis. We use a result of the simulation of glycolysis pathway of the yeast (*Saccharomyces cerevisiae*) [14]. 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 2 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 the KEGG PATHWAY data [1], where vertices represent reactions, compounds and enzymes, and edges represent relations between vertices. Usually, a reaction catalyzed by one or more enzymes contain 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. The reactions and enzymes are shown in the graph, only when all related substrates and products are activated. 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 including 101 graphs representing the glycolysis simulation for 101 seconds. DynGRL discovers 100 sets of graph rewriting rules during 100 time intervals for each threshold: 0.1 and 0.3. Then, we discovers some temporal patterns in the graph rewriting rules to describe temporal and structural aspects of the dynamic graph.

6 Results

As described in the previous section, the goal of this research is to discover temporal patterns in graph rewriting rules to describe structural changes of metabolic pathways over time. First, we show temporal patterns in graph rewriting rules. Then, we discuss structural aspects of the graph rewriting rules.

Because the modeling result represents the oscillation of glycolysis, we observe several temporal patterns in graph rewriting rules showing the oscillation. In both experiments (threshold 0.1 and 0.3), we discover oscillated temporal patterns. Using 0.1 as the threshold, temporal patterns among three chemical reactants such as C00008 (ADP), C00084 (Acetaldehyde) and C00003 (NAD^+) are discovered as shown in figure 3 (a). The points above the time axis represent the time when the substructures including each compound are removed from the graph representing the glycolysis pathway. The points below the time axis represent the time when the substructures including each compound are added to the pathway graph. For

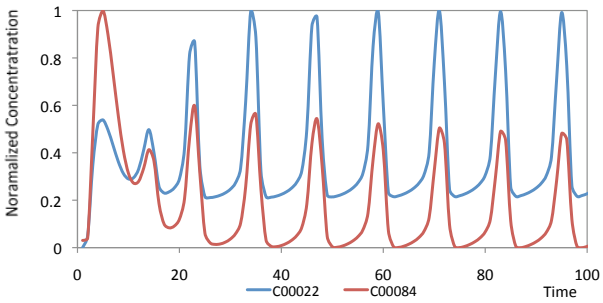

Figure 2. The oscillation curves of changing concentrations of Pyruvate (C00022) and Acetaldehyde (C00084).

example, the first red diamond time point close to 0 represent the addition of a substructure including the ADP molecule at time 2. We also observe 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 removals, they are removed following the order of ADP, Acetaldehyde and NAD^+.

Similarly, figure 3 (b) shows other temporal patterns 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 patterns with the mathematical modeling results shown in figure 2. The oscillation curves represent that Pyruvate is increased slightly earlier and decreased slightly later than Acetaldehyde. For this reason, the temporal patterns in the graph rewriting rules of our dynamic graph represent the substructures including Pyruvate are added earlier and removed later.

In the temporal aspect, the results discovered by DynGRL show several reactants in the glycolysis pathway are increased and decreased in order and these ordered patterns are repeated periodically as shown in the mathematical modeling. These patterns are observed as addition and removal of substructures of the dynamic graph representing the glycolysis pathway changing over time. The addition substructures include the increasing (over the threshold) reactants, and the removal substructures include the decreasing (under the threshold) reactants. The temporal patterns in the learned graph rewriting rules show the temporal relations that describe how the glycolysis pathway changes over time by showing which elements changes earlier than the other. These temporal patterns and graph rewriting rules help us to understand temporal properties of the glycolysis pathway.

In addition to the temporal aspect, our results can depict structural changes of the metabolic pathways. Because an advantage of the graph representation is visualization, we can understand metabolic pathways better using structural analysis as

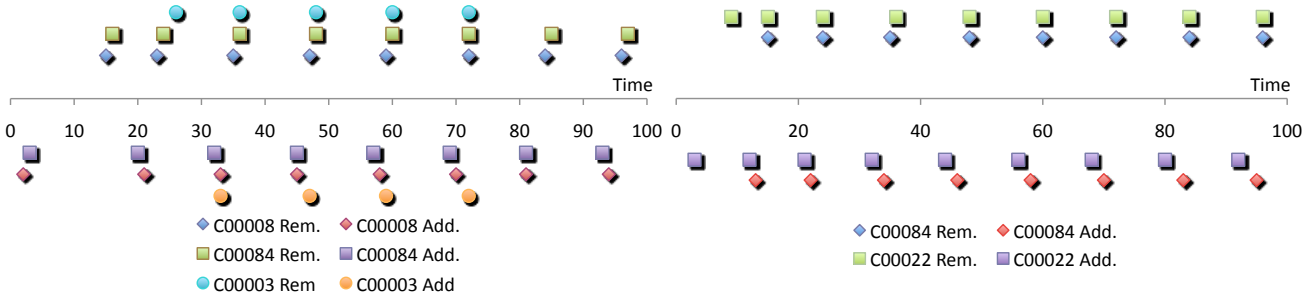

Figure 3. A visualization of time points when the substructure including each compound is removed from or added to graphs representing the glycolysis pathway at the experiment of threshold 0.1 (a) and 0.3 (b).

well as temporal analysis. Here we show some instances of substructures in the discovered graph rewriting rules, and how they are related to the original graphs.

Figure 4 shows the graphs G_i in the dynamic graph from time 24 to 32 at the experiment of threshold 0.3. These graphs show how the glycolysis pathway changes over time. In other words, what substructures are removed and added over time. The graph removes a substructure including two reactants: Acetaldehyde and Pyruvate and three related reactions, and changes its structure to the graph 25. The structure is not changed from time 25 to 31. At time 32, the graph changes more after the addition of a substructure including Pyruvate and a related reaction, R00200. The removal substructure at time 24 is also removed at time 49, 60, 72, 84 and 96. As shown in figure 3 (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 addition substructure at time 32 is also added at time 3, 12, 21, 44, 56, 68, 80, and 92 as shown in figure 3 (b). 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.

GR_k are represented as follows, ($k = 24, 25$ or $31, 32$)

$$\begin{aligned} GR_{24,25} &= \{(r_{24}, rSub1, CE_{24}, CL_{24})\} \\ GR_{31,32} &= \{(a_{32}, aSub_1, CE_{32}, CL_{32})\} \end{aligned}$$

where $rSub_1$ represents the removal substructure of G_{24} with labels marked by "-[]", and $aSub_1$ represents the addition substructure of G_{32} with labels marked by "+[]". As described in section 3, $CE_{m(n)}$ and $CL_{m(n)}$ ($m = 24$ and $n = 32$) represent sets of connection edges and connection edge labels (denoted by labels marked by "()") as follows,

$$\begin{aligned} CE_{24} &= \{(d, s2, g5), (d, s2, g9), (d, s2, g11), \\ &\quad (d, s7, g8), (d, s7, g10), (d, s7, g12)\} \\ CL_{24} &= \{S\_to\_Rct, S\_to\_Rct, Rct\_to\_P, \\ &\quad Rct\_to\_P, S\_to\_Rct, S\_to\_Rct\} \\ CE_{32} &= \{(d, s2, g5), (d, s2, g9), (d, s2, g8)\} \\ CL_{32} &= \{S\_to\_Rct, S\_to\_Rct, Rct\_to\_P\} \end{aligned}$$

The structural results show how the substructures are related to the original graphs (i.e., which connection edges link the substructures to the original graphs.) as well as what substructures are removed or added. The graph rewriting rules describe the relational aspects between the substructures (some elements) and the original graph (the pathway), but not merely which elements are changed.

In summary, our results show temporal and structural aspects of the dynamic graph representing the metabolic pathway. Temporal patterns show some elements are added and removed in order and periodically. Structural patterns show how the original graphs change after applications of removal and addition rules. These temporal patterns and graph rewriting rules help us to understand temporal properties as well as structural properties of biological networks. Some discovered temporal and structural patterns in a specific disease can show us how they are different from normal patterns and help us investigate disease and develop a new treatment.

7 Conclusion

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

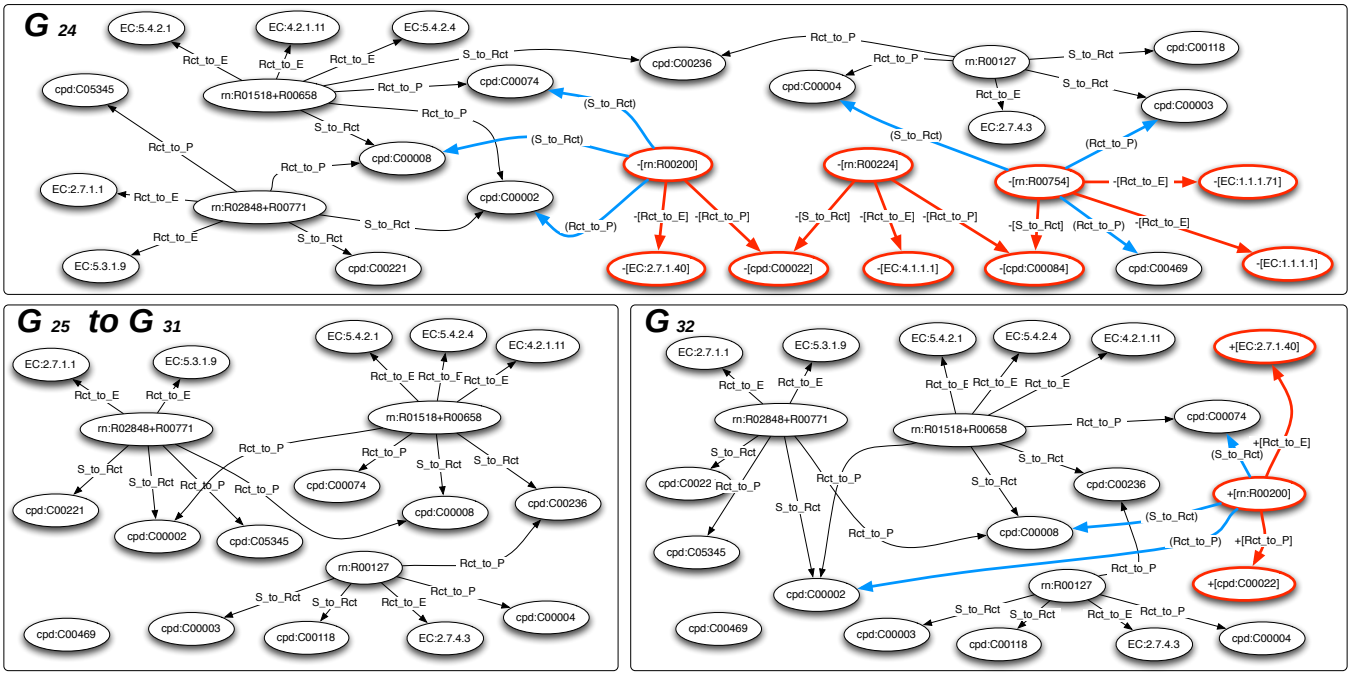

Figure 4. An example of graph transformations from time 24 to 32. Red substructure with labels marked by "-[]" or "+[]" represent removal or addition substructures. The blue edges with labels marked by "()" represent the connection edges between.

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

The future works follow several directions. We need more systematic evaluation for the learned graph rewriting rules including regenerating a dynamic graph using the learned graph rewriting rules to compare with the original dynamic graph from real world data. We will also focus on the fully automated approach to learn temporal patterns in the graph rewriting rules. Finally, we will evaluate how this approach can be used to predict future structures of biological networks using the learned patterns.

References

[1] Kyoto university bioinformatics center, KEGG website. *http://www.genome.jp/kegg/pathway*.

[2] R. Bowen. Lactose intolerance (lactase non-persistence). *Pathophysiology of the Digestive System*, 2006.

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

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

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

[6] M. Farach-Colton, Y. Huang, and J. L. L. Woolford. Discovering temporal relations in molecular pathways using protein-protein interactions. In *Proceedings of RECOMB*, 2004.

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

[8] 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.

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

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

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

[12] 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.

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

[14] 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.

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

[16] 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.

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

[18] 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.