Graph-based Temporal Mining of Metabolic Pathways with Microarray Data

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

To investigate bio-organisms and understand the theory of life, we should consider our bodies are dynamic. Our bodies are well-organized and vigorous systems, which promote reproduction and sustain our lives. Well-organized systems refer to structural properties of biological networks, which include various molecules and relationships between molecules. Vigorous systems refer to dynamic properties of biological networks, which 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.

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. 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 [3]. 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 galactose metabolism pathway after these treatments, and investigate another treatment (i.e., improving the production of the lactase enzyme in the pathway).

Temporal data mining can discover temporal features in the sequence of data. But it is hard for temporal data mining to discover structural features or relational patterns between two entities. Graph-based data mining is a process to learn novel knowledge in data represented as a graph and has been applied to identify relational patterns in biological networks [24]. However, the current graph-based data mining approaches overlook dynamic features of networks, because most of them are focused on only static graphs. 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. Our discovery algorithm takes a dynamic graph as an input. The dynamic graph contains a sequence of graphs representing biological networks changing over time. Then, the algorithm discovers rewriting rules between two sequential graphs. After discovery of whole sets of graph rewriting rules from the dynamic graph, we discover temporal patterns in the discovered graph rewriting rules.

This paper, first, introduces several preceding approaches

BIOKDD'08, August 24, 2008, Las Vegas, Nevada, USA

related to dynamic analysis of biological networks. Then, we present our definition of graph rewriting rules and our Dynamic Graph Relational Learning (DynGRL) algorithm. In our experiments, we generate several dynamic graphs of the yeast metabolic pathways using the KEGG PATHWAY database and microarray data. Then, we apply our DynGRL approach to the dynamic graphs. The results show our discovered graph rewriting rules and temporal patterns in the rewriting rules. The temporal patterns show which graph rewriting rules are repeated periodically or temporal relations among several graph rewriting rules. Our results also help us to visualize what substructures change over time and how they change. This approach enables us to investigate dynamic patterns in biological networks in two aspects: structural and temporal explorations. The ultimate goal of this research is to discover the temporal patterns in the structural changes of biological networks for drug discovery and the systems-level understanding of complex biosystems.

2. RELATED WORK

To understand how biosystems change over time, we need to follow two aspects: structural and temporal analysis of dynamic biological networks. Here, we introduce microarray analysis and temporal data mining for the temporal exploration. Then, related research on biological networks is followed for the structural exploration.

The microarray is a tool for measuring gene expression levels for thousands of genes at the same time [4, 17], 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 construct common processes or patterns in biological networks (gene regulatory networks or protein networks) in the specific condition or over time. Microarrays can also monitor patterns in gene expression levels for the period of time or at the different conditions. Patterns in gene expression levels can represent changes in the biological status or distinguish two different states, such as the normal and disease state.

Some microarray research [7, 22] describes patterns in gene expression values. One approach explores temporal patterns in gene expression promoting the regulation of a metabolic pathway [7]. Other research observes more than half of the yeast genes show periodic temporal patterns during metabolic cycles [22]. 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.

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 [20]. 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. In this research, we are focused on discovery of temporal patterns and their visualization. Allen and et al. [2] formalized temporal logic for time intervals using 13 interval relations. This approach allows us to present temporal relations in sequential data.

There are several approaches to apply temporal data mining in biological data. Ho et al. [11] 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. [9] 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 of appearance of genes such that a gene, YBR218C, appears before another gene, YGL062W, but cannot identify how these two genes interact with each other.

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. 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 relations among genes and proteins. Biological networks including metabolic pathways, protein-protein interactions and gene regulatory networks, consist of various molecules and their relationships [13]. 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 [14, 24]. 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 [18]. 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.

There are two main points to consider for understanding biological networks: structural and temporal aspects. The former reminds us to focus on relations between molecules as well as a single molecule. The latter is necessary to understand biological networks as dynamic operations rather than static relations, because every biological process changes over time and interacts with inner or outer conditions. For this reason, we need an approach to analyze biological networks changing over time in both aspects: structural and temporal properties.

3. GRAPH REWRITING RULES

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

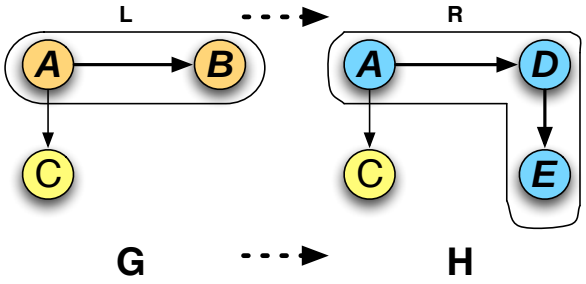

Figure 1: An example of application of graph rewriting rules, where the rule derives a graph H from a graph G by replacing a subgraph L by a subgraph R.

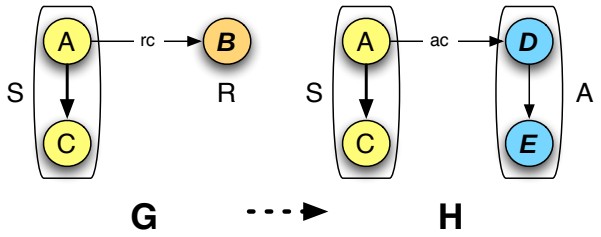

Figure 2: An example of application of graph rewriting rules, which shows an removal rule {R, rc} from a graph G and an addition rule {A,ac} to a graph H. The removal and addition substructures are connected to G and H by edges rc and ac. S represents the common subgraph between G and H.

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 [8, 21]. 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 [19]. As shown in figure 1, L is identified first in graph G. Then L is replaced by R to produce graph H. There are also several algorithms to discover the node or edge replacement graph grammar using the minimum description length principle [12, 15]. 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 2 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 rc and ac are elements of graph rewriting rules: removal and addition rules respectively. 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.

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

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 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.

We describe more detail with an example. Figure 3 shows an instance of graph rewriting rules between the synthetic biological networks, G_1 and G_2. The thick-drawn substruc-

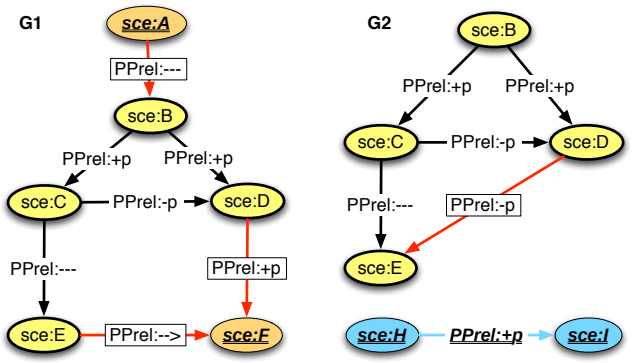

Figure 3: An instance of graph rewriting rules between graph G1 and G2 in the synthetic biological networks

tures in both graphs represent the maximum common substructures. The underline labeled elements in G_1 represent removal substructures (from G_1) with the rectangle labeled connection edges. The underline labeled elements in G_2 represent addition substructures (to G_2), where this addition rule does not have any 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 into the set 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 (indicated by '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 $\emptyset$ (emptyset) as the addition substructure, because a_1 is a rule representing a blue edge $PPrel : -p$ in G_2 without any addition substructure. a_2 also has $\emptyset$s for edges and edge labels, because $aSub_1$ represents a disconnected graph including vertices H and I in G_2.

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

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. APPROACH

This section describes our graph rewriting rule discovery system, DynGRL, that discovers graph rewriting rules in a dynamic graph. Our approach extends Cook and Holder's earlier work [5, 6], which is a graph-based relational learning approach to discover subgraphs. Their approach evaluates discovered subgraphs using the Minimum Description Length (MDL) principle to find the best subgraphs that minimize the description length of the input graph after being compressed by the subgraphs. The description length of the substructure S is represented by $DL(S)$, the description length of the input graph is $DL(G)$, and the description length of the input graph after compression is $DL(G|S)$. The approach tries to minimize the *Compression* of the graph as follows.

$$Compression = \frac{DL(S) + DL(G|S)}{DL(G)}$$

Their approach, which is called as $DiscoverSub()$ in our algorithms, tries to maximize the $Value$ of the subgraph, which is simply the inverse of the *Compression*. Even though we can use a frequent subgraph mining approach [16, 23] for $DiscoverSub()$, we choose the compression-based approach, because there is no need to choose a proper minimum support and many times the best-compressing subgraph better captures the patterns of interest than the most frequent subgraph. A more detailed comparison between the two approaches is left for future work.

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. $DiscoverSub()$ is used to find the maximum common subgraph. Although to find the maximum common subgraph is NP-Complete, $DiscoverSub()$ can be used as a polynomial-time approximation to this problem using *Limit* and *iteration* as described later. 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

Algorithm 1 DynGRL discovery Algorithm

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

1. Create $VGL = \{VG_1, VG_2, \cdots, VG_n\}$
2. $RRL = \{\}$
3. **for** $i = 1$ to $n - 1$ **do**
4. $RemRuleSet$ =AddRuleSet = $ComSubSet = \{\}$
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 if**
12. Compress $Graphs$ by $BestSub$
13. Mark $BestSub$ on VG_i and VG_{i+1}
14. **end while**
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 for**
22. **return** RRL

(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 [10]. 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.

5. DATASETS: MICROARRAY DATA AND GRAPH

We prepare dynamic graphs representing the yeast metabolic pathways in combination with microarray data. As described in section 2, microarrays can be used in two ways: monitoring the change of gene expression levels over time or distinguishing patterns in two different states. Here, we use time-based microarray data to generate a dynamic graph, where each column of data represents the gene expression values at a particular time. The microarray data used in our research observes periodic gene expression of *Saccharomyces cerevisiae* using microarray analysis [22]. The microarray data has 36 columns where each column represents one time slice. Their results show more than 50% of genes have three periodic cycles in the gene expression. We normalize each gene expression value of microarray data from 0 to 1, because we are focused on trends of the changes of gene expression values. Figure 4 shows normalized gene expression values of three genes shown in the glycolysis pathway.

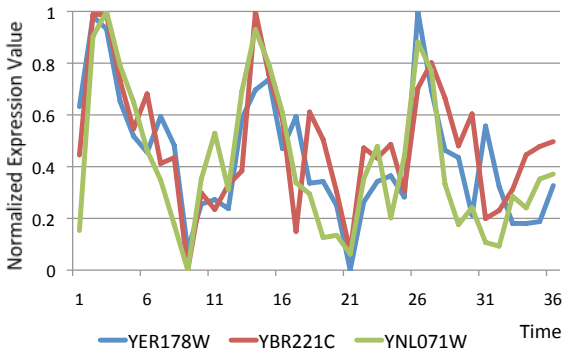

Figure 4: The oscillation curves of the changing gene expression values of three yeast genes: YNL071W, YER178W, and YBR221C.

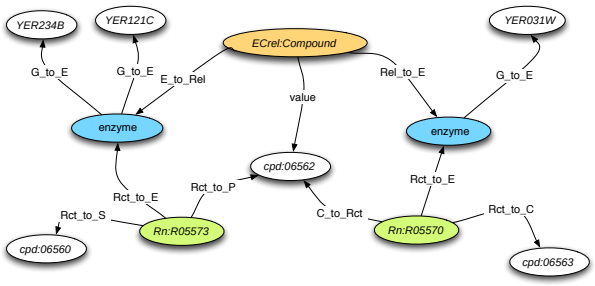

Figure 5: An instance of the graph representation for a metabolic pathway.

Here, we prepare 10 dynamic graphs, each of which contains 36 consecutive graphs representing one yeast metabolic pathway changing over time (36 time slices) corresponding to 36 columns in microarray data. The 10 dynamic graphs represent 10 metabolic pathways: glycolysis (00010), TCA (00020), Pentose phosphate pathway (00030), Purine metabolism (00230), Pyrimidine metabolism (00240), Urea cycle (00220), Glutamate metabolism (00251), Arginine and proline metabolism (00330), Glycerolipid metabolism (00561) and Glycerophospholipid metabolism (00564), where each number denotes the identification number of the pathways in the KEGG data [1]. The first three pathways are involved in the carbohydrate metabolism, the second two pathways are involved in the nucleic acids, the next three pathways are involved in the amino acids metabolism and the last two

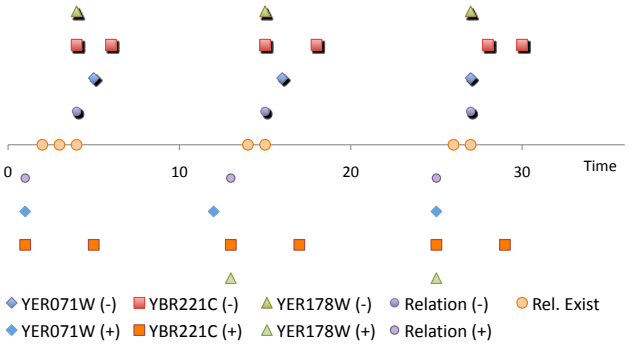

Figure 6: A visualization of time points when the substructure including each gene is removed from or added to graphs representing the glycolysis pathway at the experiment of threshold 0.6. The points above the time axis represent the time points when the substructures including the specified genes or relation are removed (Genes with (-)). The points below the time axis represent the time points when the substructures including the specified genes or relation are added (Genes with (+)). Relation points represent the time points when the enzyme-enzyme relations are shown in the pathway.

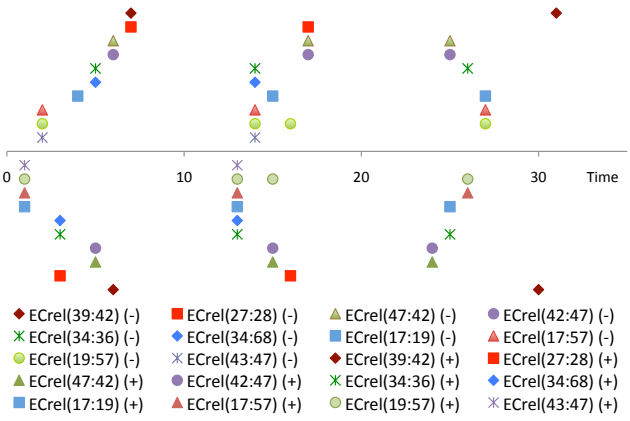

Figure 7: A visualization of time points when a particular substructure is removed from or added to graphs representing the glycolysis pathway at the experiment of threshold 0.6. Each substructure includes a relation, which is an enzyme-enzyme relation between two gene, where $ECrel(x, y)$ represents the relation, and x, y represent the id of enzymes.

pathways are involved in the lipid metabolism.

First, we generate a static graph to represent each metabolic pathway from the KEGG PATHWAY database [1], where vertices represent compounds, genes, enzymes, relations and reactions, and edges represent relationships between vertices. Figure 5 shows an example of the graph representation. "ECrel:Compound" represents a relation between two enzymes (gene products). One enzyme is produced by one or more genes, which is represented as edges "G_to_E". "RN:R$xxxxx$" represents a reaction and "cpd:C$yyyyy$" represents chemical compounds, where $xxxxx$ and $yyyyy$ represent the identification number in the KEGG database. Here, we assume only genes change over time based on gene expression values and other molecules like compounds remain the same amount.

We use a threshold t to apply the 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 the KEGG data. One enzyme needs one or more genes to synthesize. At a specific time, only one gene can be expressed out of two genes, which are needed for one enzyme. Naturally, the enzyme is not synthesized at that time. We use enzyme vertices to represent this scheme. Only when all genes are expressed, the enzyme vertex is shown in the graph. At that time, the reaction, which is catalyzed by the enzyme, is also shown. In this way, we can observe the structure of the glycolysis pathway based on microarray gene expression at each time.

6. EXPERIMENTS AND RESULTS

Our approach discovers graph rewriting rules in each dynamic graph. First, we discuss temporal patterns in graph rewriting rules. Then, we represent how the discovered substructures in the rewriting rules link to the original graphs at the specific time.

6.1 Temporal patterns

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. Because the result of the microarray data [22] represents three periodic cycles of gene expression, we observe similar temporal patterns in graph rewriting rules. Here, we are focused on graph rewriting rules involving enzyme-enzyme relations as well as genes. The enzyme-enzyme relation represents a relationship between two enzymes. As shown in figure 5, one or more genes produce an enzyme, and the enzyme can have a relation with one other enzyme. The relation vertex labeled as "ECrel:Compound" exists, only when there exist two enzyme vertices. Each enzyme vertex exists only when the linked genes exist (biologically, the linked genes produce the enzyme). The left enzyme exists only when two genes, YER178W and YER221C exist. The right enzyme exists only when one gene YAL038W exists.

Figure 6 shows a visualization of the changes to the partial pathway including the above three genes of the glycolysis pathway. The complete pathway is shown in figure 10 (Sub F). 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. The points on the axis represent the time when the relation exists. The result clearly shows the temporal patterns in removal and addition rules as three cycles. Three genes are added and the relation is shown in the pathway. After several time intervals, one of three genes starts to be removed from the pathway and the relation disappears, too. Like the microarray research [22], we can notice the genes are added and removed three times periodically. In addition, we discover the removal and

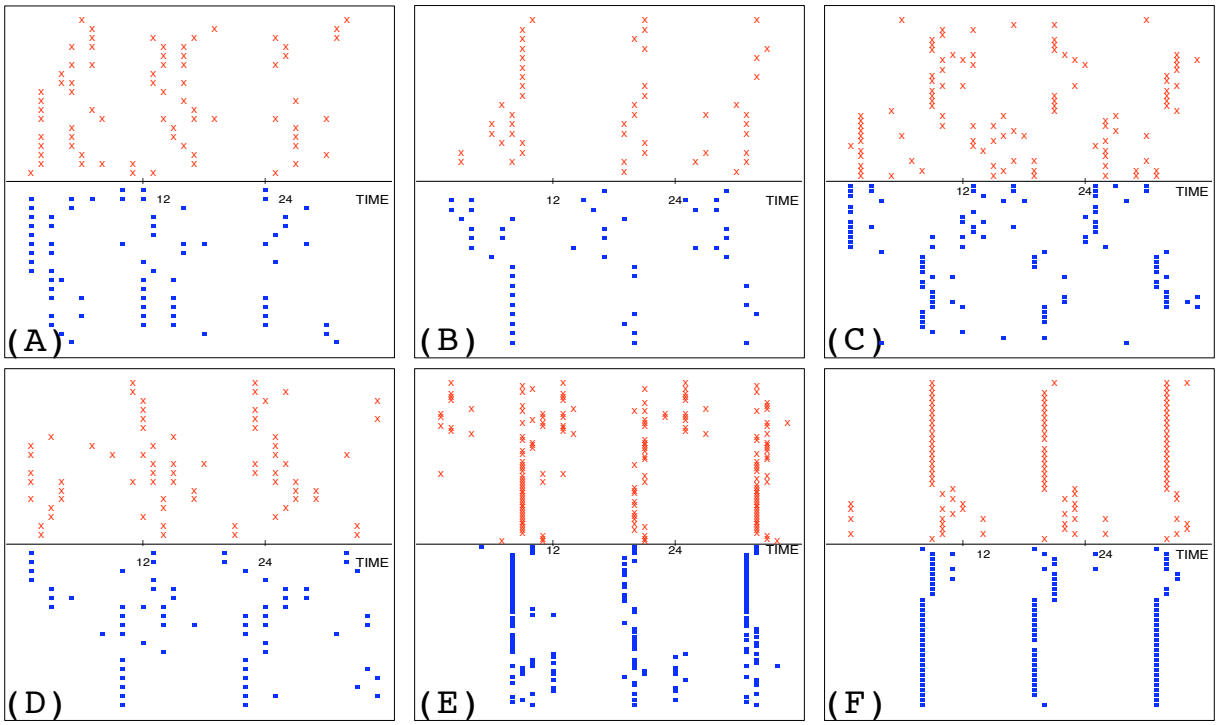

Figure 8: Visualization of three periodic cycles in removals and additions of Enzyme Relations in TCA cycle (A), urea cycle (B), glutamate metabolism (C), glycerophospholipid metabolism (D), purine metabolism (E) and pyrimidine metabolism (F) at the experiment of threshold 0.5. The points marked as "X" above the time axis represent removals, and the points marked as rectangles represent additions.

Table 1: Running time of ten dynamic graphs. Pathway denotes the name of the pathway represented by the dynamic graph. Max. Size and Min. Size denote the maximum and minimum size of a graph in the dynamic graph. Total Size denotes $\sum size(G_i)$ for $G_i \in DG$. Time is in seconds

Pathway	Max. Size	Min. Size	Total Size	Time
00010	522	65	7738	69.86
00020	294	46	4667	9.44
00030	192	57	4069	3.82
00220	236	58	4147	4.58
00251	394	110	7928	172.88
00330	184	61	4277	4.65
00561	183	44	2425	3.38
00564	231	57	4937	4.96
00230	643	161	10259	54.06
00240	486	85	6040	18.03

addition of some relations also show temporal cycles. Suppose there are two genes and a relation between two genes. One gene is always shown in the pathway, and the other is shown three times periodically. The relation is also shown three times like the latter gene, because the relation is activated only when both genes are activated. Because most genes and proteins work together, the temporal patterns in the relations between the molecules are also important as well as the temporal patterns in the existence of genes and proteins.

Figure 7 shows a visualization of three periodic cycles of 10 relations in the glycolysis pathway. In this experiment, the dynamic graph with threshold 0.6 shows a maximum of 13 relations at each time slice. 10 out of the 13 relations clearly show periodic cycles three times. Figure 8 shows the similar temporal patterns in the six other pathways, TCA cycle (A), urea cycle (B), glutamate metabolism (C), glycerophospholipid metabolism (D), purine metabolism (E) and pyrimidine metabolism (F). The points (marked as "X") above the time axis represent the patterns of removals and the points (marked as the rectangles) below the time axis represent the patterns of additions. The two time points with the same distance over the axis represent the removals and additions of the same subgraphs. The six visualizations show the temporal patterns in the graph rewriting rules of the major metabolic pathways. Even though there are some time points that do not show clear cycles, all ten pathways show the three periodic cycles of enzyme-enzyme relations. We can conclude that the removals and additions of the subgraphs including genes and relations show the temporal patterns of three periodic cycles. Table 1 shows the running time of Algorithm 1 on the ten dynamic graphs representing the ten metabolic pathways. Most cases are finished within a minute.

Figure 9 shows the temporal patterns in maplink-relations, which represent the relations between two enzymes that belong to two different pathways. $Link(+)$ denotes the time points when two pathways are linked to each other, and $Link(-)$ denotes the time points when they are disconnected. Because these relations are also activated by the

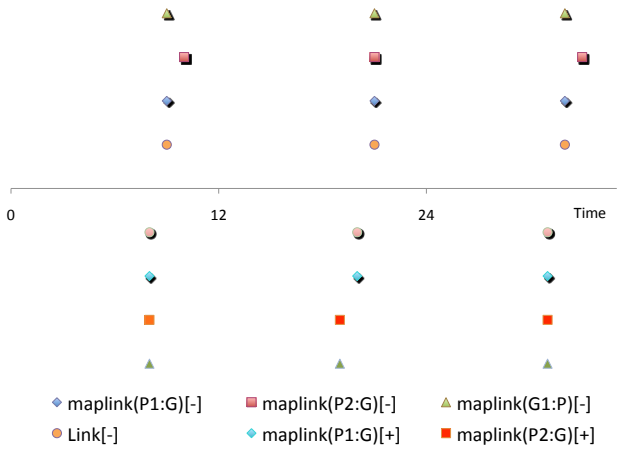

Figure 9: A visualization of three periodic cycles in time points when two pathways (Purine metabolism (00230) and Glutamate metabolism (00251)) are linked to each other at the experiment of threshold 0.5.

gene expression values, they also show three periodic cycles like enzyme-relations. In fact, all metabolic pathways in a cell are connected to each other. Practically, we classify the metabolic pathways for each function such as glycolysis, urea cycle and so on. The temporal patterns in the maplink-relations show when two pathways are connected or disconnected to each other. In addition to the temporal patterns, our results show the structural properties related to these patterns in next section.

Our results show three periodic cycles of enzyme-relations and maplink-relations over ten major metabolic pathways. We can observe similar temporal patterns in the four major categories of pathways. These temporal patterns of relations describe periodic cycles in the behaviors of the yeast biosystem corresponding to the periodic cycles of the gene expression of the yeast. The major events and behaviors of the biosystems accord with the metabolic cycles [22].

The experiments show that DynGRL discovers graph rewriting rules from dynamic graphs representing the yeast metabolic pathways changing over time. These graph rewriting rules represent temporal patterns that describe how the structure of the metabolic pathways change over time by showing which elements change periodically. These temporal patterns and graph rewriting rules help us to understand dynamic properties of the metabolic pathways. The results show not only temporal patterns in structural changes of metabolic pathways, but also temporal patterns in the connections between two different pathways.

6.2 Structural patterns

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

Figure 10 shows structural changes of the dynamic graph representing the partial glycolysis pathway introduced in figure 6. G_i represents the graph at time i. This dynamic graph contains 36 time series of graphs starting with a single vertex graph in time 1 to no vertex in time 36. The blue edge with the boxed labels between two sequential graphs represents the graph transformation using removal (-) or addition (+) of one of the six substructures (Sub A to F). For example, graph G_5 is transformed to G_6 with removal of Sub C and addition of Sub B. The red edges with the dot boxed labels in the rules represent the connection edges as described previously. The connection edges describe how the discovered substructures connect to the original graph.

As described previously, we show the graph rewriting rules between two graphs as a formula. Here, we show two examples of graph rewriting rules $GR_{1,2}$ and $GR_{5,6}$ as follows,

$$\begin{aligned} GR_{1,2} &= \{a_1, add_A, CE, CL)\}, \\ &\quad CE = \{(d, S2, G2)\},\ CL = \{G\_to\_E\} \\ GR_{5,6} &= \{(r_1, rem_C, \emptyset, \emptyset), (a_1, add_B, \emptyset, \emptyset)\} \end{aligned}$$

where a_m and r_n denote the indices of the removal and addition rule in each graph rewriting rule, add_x and rem_y denote the substructure (Sub A to F) in figure 10. CE and CL denote the connection edges and connection edge labels respectively. The connection edge with a label G_to_E links Sub A to a gene YER178W in G_1 so that an enzyme is activated by two genes, YBR221C and YER178W, and a relation is created with the other enzyme that is activated by a gene, YNL071W. But CE and CL are all $\emptyset$ in $GR_{5,6}$ because there is no connection edge between the substructures (rem_C and add_B) and the original graphs (G_5 and G_6) respectively.

Figure 11 shows our visualization results of a removal and addition rule. The left figure shows a removal rule in our output and the right figure shows the same rule marked on the KEGG pathway map. The labels marked by "-[]" represent the labeled vertices and edges belonging to the substructures of removal rules. The labels are marked by "+[]" in the case of addition rules. Connection edges between the discovered substructures and original graphs are marked by "()". The removal of a gene YKL060C causes the removal of two enzyme-relations with one other gene YDR050C and a reaction R01070, which is a catalyzed by an enzyme produced by YKL060C (There can exist more than one relation with different properties between two genes in the KEGG data.). The graph also loses several connection edges between the removal structures and original graph. The DynGRL system helps us visualize removal or addition rules on the original graph with the connection edges. The results show how the substructures in graph rewriting rules are structurally connected to the original graphs and how the graphs change after removal or addition rules are applied.

In addition to the change of one element, our results show how the changes are related to other elements (i.e., which elements are removed or added at the same time) as shown in the discovered subgraphs and how the subgraphs are linked to the original graphs. Our results show patterns in the structural changes, not merely changes of amount. It allows us to better understand the structural properties as the pathways change over time.

In summary, we evaluated our algorithm in the experiments with 10 dynamic graphs each containing 36 graphs representing the yeast metabolic pathways in combination with the microarray data of yeast. 35 sets of graph rewrit-

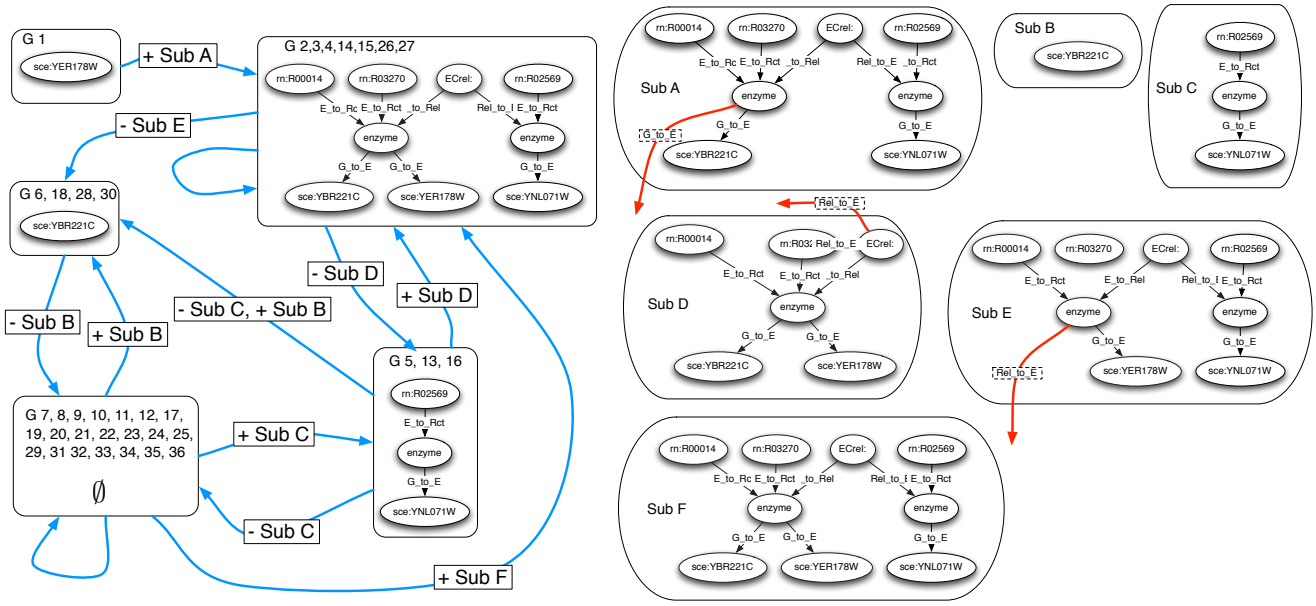

Figure 10: Structural changes of a dynamic graph representing the partial glycolysis pathway. Gi denotes a graph at time i for $1 \leq i \leq 36$. The blue arrows with boxed labels between two graphs, Gx and Gy, represent the transformation from Gx to Gy by application of the rule in the label of the arrow. Sub p (A to F) represents the substructure in each rule (removal and addition), where the red arrows with the dot boxed labels from the substructures represent the connection edges. For example, $G1$ is transformed to $G2$ by addition of Sub A, which is connected by a connection edge labeled "G_to_E".

ing rules for removals and additions are discovered during 35 time intervals. Temporal patterns in the graph rewriting rules show a number of substructures are removed and added periodically as showing three cycles. The graph rewriting rules and our visualization results describe how the discovered substructures are connected to the original graph and how the structures of graphs change over time. 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

This research formalizes graph rewriting rules to describe structurally changing biological networks and proposes an algorithm, DynGRL, to discover graph rewriting rules in a dynamic graph. The algorithm is evaluated with the dynamic graphs representing the yeast metabolic pathways in combination with the microarray data. Our approach represents structural and temporal properties at the same time, and discovers novel patterns in both properties. The results show our dynamic graph-based relational learning approach discovers several novel temporal patterns in graph rewriting rules of the metabolic pathways such that some relations between genes and pathways are shown periodically. Additionally, the results show periodic cycles of temporal patterns in connections between two pathways. DynGRL can also help us to visualize the removed or added substructures to show how the graphs structurally change or how the substructures in rewriting rules are related to the original graphs.

The graph rewriting rules of biological networks can describe how the complex biosystems change over time. The learned temporal patterns in the rewriting rules can describe not only structural changes of metabolic pathways but also temporal patterns in series of the structural changes. Our approaches help us to better explore how biological networks change over time and guide us to understand the structural behaviors of the complex biosystems. Specifically, the temporal patterns in structural changes of the biosystems under specific conditions (e.g., infection) can provide essential information for drug discovery or disease treatment.

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

8. REFERENCES

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

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

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

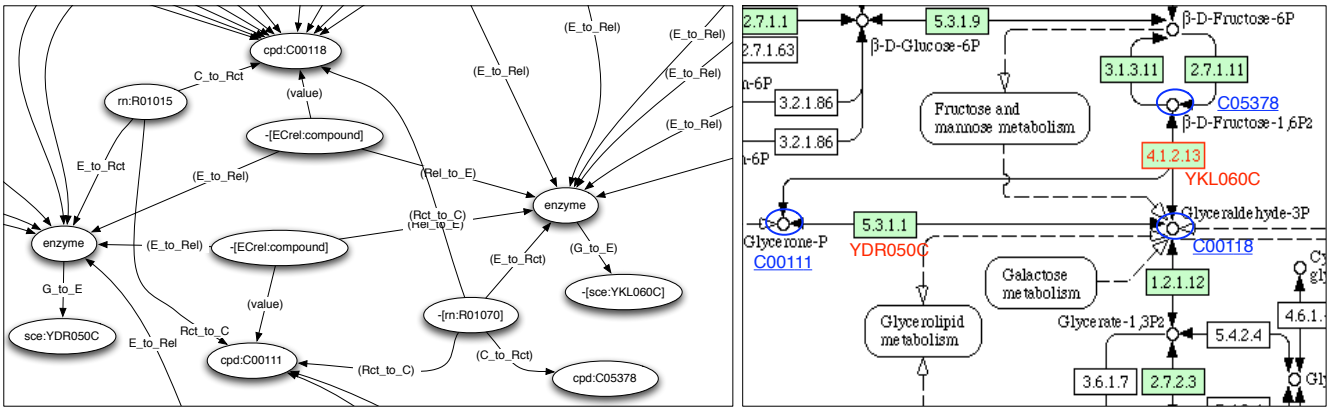

Figure 11: A visualization of discovered substructures of removal rule in a dynamic graph representing the glycolysis pathway in our output (left) and on the KEGG glycolysis pathway map (right). Labels marked by "-[]" represent the removal rules and labels marked by "()" represent the connection edges (left). Red rectangles represent two genes and blue circles represent three compounds in the removal rule (right). When YKL060C is removed, two enzyme-relations between two genes are also removed. Two reactions R01015 and R01070, involved with the three compounds, are also removed.

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

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

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

[7] J. DeRisi, V. Iyer, and P. Brown. Exploring the metabolic and genetic control of gene expression on a genomic scale. *Science*, 278(5338):680–686, October 1997.

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

[9] 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.

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

[11] 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.

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

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

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

[15] J. P. Kukluk, L. B. Holder, and D. J. Cook. Inference of node replacement recursive graph grammars. In *Proceedings of the SDM*, 2006.

[16] M. Kuramochi and G. Karypis. Frequent subgraph discovery. In *Proceedings of the ICDM*.

[17] D. J. Lockhart and E. A. Winzeler. Genomics, gene expression and dna arrays. *Nature*, 405:827– 836, 2000.

[18] 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.

[19] 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.

[20] 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.

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

[22] B. Tu, A. Kudlicki, M. Rowicka, and S. McKnight. Logic of the yeast metabolic cycle: Temporal compartmentalization of cellular processes. *Science*, 310:1152–1158, 2005.

[23] X. Yan and J. Han. gspan: Graph-based substructure pattern mining. In *Proceedings of the ICDM*.

[24] 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.