

# Side Effect Prediction using Cooperative Pathways

Mutsumi Fukuzaki, Mio Seki  
Department of Computer Science,  
Ochanomizu University,  
Tokyo, Japan.  
{fukuzaki,seki}@sel.is.ocha.ac.jp

Hisashi Kashima  
Tokyo Research Laboratory,  
IBM Research,  
Kanagawa, Japan.  
hkashima@jp.ibm.com

Jun Sese  
Department of Computer Science,  
Ochanomizu University,  
Tokyo, Japan.  
sesejun@is.ocha.ac.jp

**Abstract**—Drugs and biological experiments are designed to affect a particular target gene or pathway. However, they might inadvertently activate other pathways and cause side effects. Because of the existence of complex cellular mechanisms responding to stimuli, it is difficult to detect the presence of such side effects. Therefore, identification of pathways that function together under identical conditions would greatly help in anticipating these side effects before conducting these experiments.

We develop a novel method to enumerate “cooperative pathways” defined as pathways that function together under identical conditions by combining pathway networks with comprehensive gene expression profiles. For finding cooperative pathways from whole pathways, we propose an efficient algorithm, CoopeRativE Pathway Enumerator (*CREPE*), which enumerates connected subpathways having common activate conditions and selects combinations of the subpathways sharing the conditions.

We apply *CREPE* to a yeast stress dataset combined with the KEGG pathways. We observe that the starch and sucrose metabolism pathway cooperates with the pyruvate metabolism under heat shock stresses. It cooperates with the tricarboxylic acid (TCA) cycle under the stationary phases.

**Keywords**-gene expression; pathway; active network; itemset

## I. INTRODUCTION

Most biological experiments and those involving drugs are designed to affect a single target gene or a pathway. However, they frequently result in various changes in non-target genes or pathways within a cell. These unexpected effects are termed side effects; and they might engender undesirable pathological changes and interfere with the experimental results: therefore, it is important to detect these side effects. However, it is difficult to predict their existence because currently, little information is available about the mechanisms especially in higher organisms, and most such unknown side effects occur only under specific conditions or when specific drugs are used.

Therefore, it would be beneficial to identify and enlist all pathways that function together under identical experimental conditions. To find the set of pathways and conditions, we

This work was partially supported by KAKENHI (Grant-in-Aid for Scientific Research) on Priority Areas “Systems Genomics” from the Ministry of Education, Culture, Sports, Science and Technology of Japan, and a research grant from Leave a nest Co., Ltd. H.K. is currently with Department of Mathematical Informatics, the University of Tokyo.

combine known pathway networks with gene expression profiles. We develop a novel method called the CoopeRativE Pathway Enumerator (*CREPE*), which simultaneously handles pathway networks and gene expression profiles and enumerates subnetworks that share conditions in which the genes are overexpressed or suppressed.

In this study, we design a model to enlist the side effects of drugs by studying the target pathways and their corresponding gene expression profiles. A pathway is represented as a graph, in which a vertex indicates a gene or a chemical compound and an edge indicates a chemical reaction or a gene interaction between genes or compounds.

Figure 1(A) presents an example of the pathway network to which we refer as  $G$ . For ease of presentation, we represent pathways as undirected graphs in this paper. But, our method is applicable for directed and undirected graphs. Each vertex  $v_i$  denoting a gene is associated with a set of conditions representing the set of drugs or conditions activating that vertex. We call the set of drugs or conditions an *itemset*. For example, vertex  $v_0$  is associated with an itemset  $\{i_1, i_3\}$ , indicating that drugs or conditions  $i_1$  and  $i_3$  act on  $v_0$ . We will determine the activation environments from gene expressions.

Based on this network, we seek multiple sub-pathways sharing common activation drugs because the existence of such cooperative pathways implies hidden or hitherto unknown connections among the pathways and that drugs designed to target genes in a pathway might also act on other pathways.

Figure 1(B) depicts an example of the sub-pathways which are shown as bold lines. The subnetwork, consisting of three vertices  $v_1$ ,  $v_4$ , and  $v_5$ , has a common itemset  $\{i_1, i_2\}$  shared by the three vertices; the other subnetwork consisting of  $v_3$ ,  $v_7$  and  $v_9$  also shares the common itemset  $\{i_1, i_2\}$ . These two subnetworks show side effects because they share the activation conditions  $\{i_1, i_2\}$ . We call the pathways “cooperative pathways” (*CP*). The conditions, designed to target one of the two sub-pathways, might cause activations or changes in the other sub-pathway. In the example presented above,  $v_8$  also contains the itemset  $\{i_1, i_2\}$ . However, no vertex adjoining  $v_8$  contains  $\{i_1, i_2\}$ ; therefore, activation would result in this case from accidental expression observations. A larger subnetwork would be more

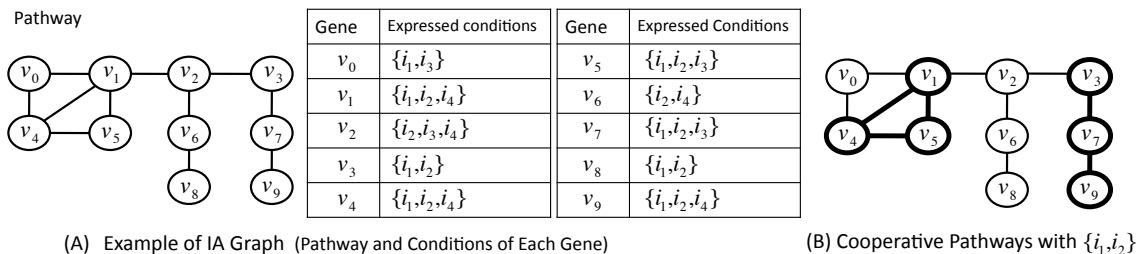


Figure 1. An example of a frequently occurring network with pathways sharing common conditions

reliable; prediction of the side effects would be of greater concern because they would cover a wider range of pathway networks. Furthermore, we can expect that the larger the common itemset, the more probable are the side effects. Therefore, the more probable CP consists of the larger networks and larger size of itemsets. Therefore, CP consisting of the large sub-pathways with large common activation conditions would be useful for drug and experimental design and prediction of side effects.

In this paper, we formalize the problem of mining CPs, implying unknown connections between pathways and drug side effects. To solve the problem, we introduce a novel method called CoopeRativE Pathway Enumerator (*CREPE*), which simultaneously handles pathway networks and itemsets. Using this technique, we discover that starch and sucrose metabolism cooperates with pyruvate metabolism under heat shock stress in yeast, whereas it cooperates with the tricarboxylic acid (TCA) cycle in the stationary phases.

Mining cooperative pathways is associated with graph mining researches. Among its related topics, frequent subgraph discovery methods [1]–[3] enable us to enumerate all subgraphs whose number of occurrences is greater than a user-specified threshold, and have been applied to chemical compound mining. In this study, we are unconcerned with the subgraph structure. In addition, the existing methods cannot handle itemsets on the subgraphs. Therefore, we can not apply the existing methods directly to our problem.

One straightforward approach for finding CPs might be the use of frequent pattern mining methods [4], [5] to obtain all frequent itemsets and then determine their interconnections in the networks. However, because numerous candidate itemsets would be generated in the first stage, this assessment would be very time consuming. In our study, we design a tailored algorithm to enumerate subgraphs rapidly using common itemsets. Therefore, we first generate candidate subgraphs and then assessed combinations of the subgraphs.

Combinatorial mining of networks with numerical gene expressions has been studied in constrained clustering [6], [7]. The studies attempt to identify the simultaneous clustering of the vertices in a network and their associated numerical vectors. One major difference between these analyses and ours is that in our study, the associated features on

every vertex are discrete values. This property makes it difficult to apply the existing constrained clustering methods to our problem. MATISSE [8] and CoPaM [9] execute the combinatorial mining of networks with feature vectors. Both methods find dense subgraphs whose vertices having similar features. However, we are unconcerned with the subgraph density; our method can find the *sparse* hub network shown in Figure 4(A). We will compare our result with MATISSE result in Section III.

Hashimoto *et al.* [10] proposed a combinatorial mining of sequence structured data and tree structured data. Their approach can be similarly applied to graph structured data, but the goal of our problem is not enumerating frequent subgraphs. In addition, Seki and Sese [11] introduced a problem to find the largest *connected* common pattern graph. In this paper, we specifically examine enumeration of frequent *disconnected* graphs. To associate gene expressions with pathways, gene set enrichment analysis [12] has been used. The method can select the most associated pathway from a single gene expression profile. However, the method can not predict exact combinations of pathways working together implying side effects.

## II. METHODS

### A. Preliminaries

In this section, we introduce a novel data mining problem for analyzing itemset-associated graphs, which we refer to as the ISS set enumeration problem. Then, we select CPs from the ISS sets.

Let  $G$  be an undirected, unlabelled, and unweighted graph with an itemset on each vertex. We refer to this graph as an *itemset-associated graph (IA graph)*. Let  $V(G)$ ,  $E(G)$  and  $\mathcal{I}(G)$  respectively signify a set of the vertices in  $G$ , a set of edges in  $G$  and a set of itemsets on vertices in  $G$ . Note that the size of graph  $G$  is given as the number of edges, i.e.,  $|G| = |E(G)|$ . Each vertex in  $G$  has an itemset. Let  $I(v)$  be the itemset on  $v \in V(G)$ .

We next define subgraphs whose vertices share itemsets.

*Definition 1:* (Shared Itemset) Let  $G'$  be a connected subgraph of an IA graph  $G$ , where  $G'$  is also an IA graph. We define  $I(G')$  as  $I(G') = \bigcap_{v \in V(G')} I(v)$ , and refer to  $I(G')$  as a *shared itemset* of  $G'$ .

Among the subgraphs having a shared itemset, we focus on an important subset, which cannot be expanded while retaining the currently shared itemsets.

**Definition 2:** (Itemset-Sharing Subgraph (ISS)) We call  $G'$  an *itemset-sharing subgraph (ISS)* with  $I(G')$  if  $I(G') \neq \phi$  and  $I(v) \not\supseteq I(G')$  for any vertex  $v$  in the neighbor vertices of  $G'$ .

Now, we define the sets of ISSes that we want to enumerate in our task. As described in Section I sets of ISSes are useful to find side effects of drugs.

**Definition 3:** (ISS Set) Let  $\mathcal{G} = \{G_1, G_2, \dots, G_n\}$  be a set of ISSes, where each  $G_i$  is an ISS. Define  $I(\mathcal{G})$  as  $I(\mathcal{G}) = \bigcap_{G \in \mathcal{G}} I(G)$ . Note that  $I(\mathcal{G}) = \bigcap_{G \in \mathcal{G}} \bigcap_{v \in G} I(v)$ . We call  $\mathcal{G}$  an *ISS set* with  $I(\mathcal{G})$ , if all of the following conditions are satisfied:

- 1)  $V(G_i) \cap V(G_j) = \phi$  for any  $G_i$  and  $G_j$  ( $i \neq j$ ) in  $\mathcal{G}$ .
- 2)  $I(v) \not\supseteq I(\mathcal{G})$  for any vertex  $v$  in the neighbor vertices of  $G' \in \mathcal{G}$ .
- 3)  $|G_i| \geq \theta_S$ , where  $\theta_S$  is a user-specified value.
- 4) No ISS  $G'$  with  $|I(\mathcal{G})|$  exists except in  $\mathcal{G}$ .

The first two conditions are an extension of the definition of ISS for dealing with multiple ISSes. The third condition gives the minimum size of the obtained ISSes because larger ISSes are of greater interest to us. The last condition ensures the maximality of the found ISS sets.

Let  $|\mathcal{G}|$  indicate the number of disconnected components of  $G$ , and hence,  $|\mathcal{G}| = n$ .

Finally, we define our new data mining problem where the task is to enumerate all ISS sets from a given IA graph.

**Definition 4:** (ISS Set Enumeration Problem) Given an IA graph and user-specified values  $\theta_S$ ,  $\theta_I$  and  $\theta_F$ , enumerate all ISS sets  $\mathbf{G}$  satisfying

- $|\mathcal{G}| \geq \theta_F$ ,  $|I(\mathcal{G})| \geq \theta_I$  for any ISS set  $\mathcal{G} \in \mathbf{G}$ , and
- $|G| \geq \theta_S$  for any ISS  $G \in \mathcal{G}$  in any ISS set  $\mathcal{G} \in \mathbf{G}$

from the IA graph.

## B. Enumeration of CPs

In this section, we propose an efficient algorithm called *CREPE* (CoopeRativE Pathway Enumerator) for solving the ISS set enumeration problem. CREPE consists of three stages. In the first stage, we enumerate all the ISSes efficiently by introducing *DFS itemset tree*. In the second stage, we generate ISS sets by combining the ISSes with avoiding the generation of numerous combinations of ISSes. In the last stage, we select larger and non-overlapped ISS sets from the ISS sets. The selected ISS sets are CPs.

1) *ISS enumeration:* In the first stage of the CREPE, we enumerate ISSes. We introduce efficient techniques for the enumeration of ISSes in this section.

We use a depth-first search (DFS) tree for enumerating ISSes  $\mathcal{G}$  where  $|G| \geq \theta_S$  and  $|I(\mathcal{G})| \geq \theta_I$  for  $G \in \mathcal{G}$ . Each node of the tree contains a vertex and an itemset related to the path from the root to the node. We denote the tree as a *DFS itemset tree*. On the DFS itemset tree, we do not

need to maintain edges because  $I(G)$  can be computed from vertices and their itemsets.

The generation of the subgraphs itself is considered to be a simplified version of the DFS lexicographic order used in the gSpan algorithm [3], and hence, this DFS itemset tree can avoid duplicate generation of identical graphs.

Figure 2 shows the DFS itemset tree for the IA graph in Figure 1(A). Each node in the DFS itemset tree contains a vertex and an itemset. Each tree node corresponds to a subgraph, and its associated itemset indicates the shared itemset of the subgraph. The vertices included in the path from the root to the tree node represent the vertices of the subgraph.

Thanks to the following monotonic property of ISS about itemset size, we can prune subtrees in the DFS itemset tree, which dramatically reduces the search space.

**Property 1:** Let us denote two ISSes by  $G'$  and  $G''$ , and let  $V(G') \supset V(G'')$ . Then,  $I(G') \subseteq I(G'')$  holds.

The tree nodes indicated by dotted boxes in Figure 2 can be pruned by using this property.

Using the DFS itemset tree, we can generate all ISSes whose subgraph size is greater than  $\theta_S$  and common itemset size is greater than  $\theta_I$ . Figure 3(A) illustrates the ISSes and their associated itemsets. We refer to this table as the *ISS table*.

2) *ISS set combination:* In this section, we introduce an efficient method for enumerating ISS sets from the ISS table created in the previous section.

Once we fix one itemset, an ISS set sharing the itemset is uniquely determined. Therefore, one approach to enumerating all ISS sets is to generate all itemsets that can be associated with ISS sets. For the efficient generation of the itemsets, we use the depth first search to enumerate all of the combinations of ISSes.

**Definition 5:** (ISS Tree) Let  $T_I$  be a tree, each of whose node  $n$  contains itemset  $I(n)$  and a set of ISSes  $\mathcal{G}(n)$  which shares  $I(n)$ . The root of  $T_I$  contains an itemset including all items and a vacant set of ISSes. Let  $n_1$  and  $n_2$  be a pair of nodes of  $T_I$ . When  $n_1$  is an ascendant of  $n_2$ ,  $I(n_1) \supseteq I(n_2)$ .

We call the tree *ISS tree*. Nodes closer to the root contain larger itemset.

Figure 3(B) shows the ISS tree. Each node contains an ISS set, and the root node contains all the items and an empty set of ISSes.

We can prune the branches in the ISS tree from the monotonic property in Definition 5. Furthermore, the following property substantially reduces the search space.

**Property 2:** Let node  $n$  contain an itemset  $I(n)$  and an ISS set  $\mathcal{G}(n)$ . If  $I(n)$  and an itemset  $I'$  of an existing node are identical, we need not traverse the branch rooted by  $n$ .

In Figure 3(B), the nodes with cross signs indicate that the nodes are pruned since their itemset size is less than the threshold. The gray nodes were pruned by Property 2. To use these pruning techniques, we need not calculate inclusion

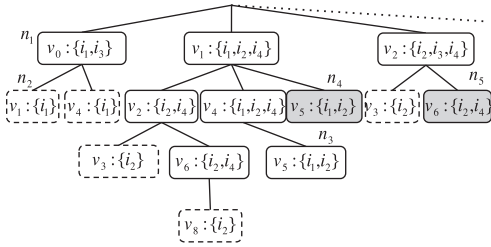


Figure 2. The DFS itemset tree for Figure 1(A)

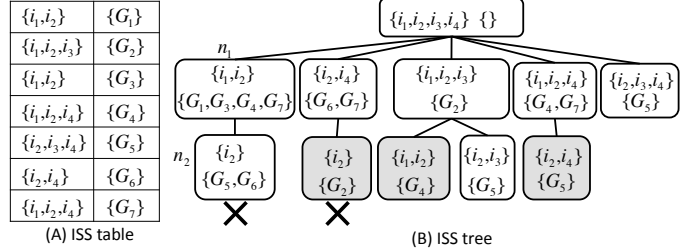


Figure 3. An ISS table and an ISS tree

relations between graphs in ISSes. The other nodes with no less than  $\theta_F$  ISSes in the ISS tree contain ISS sets.

3) *Selecting CPs*: The enumerated ISS sets might include graphs with several overlaps of edges and experimental conditions. Overlapped subgraphs are frequently caused by items which highly correlate with each other. To select disconnected subgraphs from the enumerated ISS sets, we introduce the following heuristics.

Let  $\mathcal{G}_C$  and  $G_p$  be a set of ISS sets and a (disconnected) graph, where  $\mathcal{G}_C$  is a candidate of a selected ISS set, and  $G_p$  is the set of IA-graphs checked already. We would like to select an ISS set which do not overlap with  $G_p$ . We define *overlapping ratio*  $overlap(\mathcal{G}_C, G_p)$  as

$$\left( \prod_{G \in \mathcal{G}_C} \frac{|E(G) \cap E(G_p)|}{\min\{|E(G)|, |E(G_p)|\}} \right) \times \frac{|I(\mathcal{G}_C) \cap I(G_p)|}{\min\{|I(\mathcal{G}_C)|, |I(G_p)|\}},$$

where  $|E(G)|$  and  $|I(G)|$  are the number of edges in  $G$  and the itemset size of  $I(G)$ , respectively. When  $\mathcal{G}_C$  and  $G_p$  have no common edges or samples,  $overlap(\mathcal{G}_C, G_p)$  is zero. On the other hand, when  $\mathcal{G}_C$  is included in  $G_p$ ,  $overlap(\mathcal{G}_C, G_p)$  is close to one. Using a user-specified threshold of the overlapped ratio  $r$ , we identify subgraphs in  $\mathcal{G}_C$  overlapping with  $G_p$  by the rule  $overlap(\mathcal{G}_C, G_p) > r$ . Using this heuristics, we can identify important subgraphs to which we refer CPs.

### III. RESULTS

In this section, we demonstrate the results of applying CREPE to the yeast stress dataset with the KEGG pathways, and discuss their biological significance.

#### A. Pathway network data

We obtain gene interaction data from the KEGG pathways [13]. Although the pathway contains proteins and chemical compounds, all chemical compounds are excluded from the network because none is related to the gene expressions. To avoid disruption in the pathway network that occurs by omitting the compounds, we generate edges between every gene connected via chemical compounds. We combine all the pathways. Therefore, different pathways are connected if one gene is involved in multiple pathways. All the pathway datasets were downloaded on Dec. 18, 2008.

#### B. Yeast stress dataset

1) *Preparation*: We use a yeast stress dataset to investigate whether or not CREPE can accurately determine side effect networks. This dataset contains the gene expressions of 6,152 genes under 173 different stress environments [14].

We first convert numerical gene expression data into a set of over-expressed conditions. When the expression value of gene  $g$  under condition  $c$  is no less than user-specified threshold  $\theta_E$ , then  $g$  is considered highly expressed under  $c$ . We set the value of  $\theta_E$  as 1.5, in which the average number of overexpressed environments over all genes is 4.84 (approximately 2.5% of all environments). We also generate a suppressed dataset. When the expression is less than  $\theta_E$ , we regard the gene as suppressed. We set  $\theta_E$  as  $-1.7$ , and its average number of suppressed environments is approximately 2.5%.

In CREPE, the following three values are user specified:  $\theta_F$ ,  $\theta_I$ , and  $\theta_S$ . In order to find disconnected subgraphs causing side effects, we set  $\theta_F = 2$ . In the yeast expression profile, conditions of 173 types are classified into 32 groups. More than five similar experiments were performed for each group (time course or temperature-dependent samples). To detect conditions expressed in at least more than half of the environments, we set  $\theta_I = 3$ . When  $\theta_S$  is small, many false-positive pathways might be enumerated in the case in which two genes directly connected in a pathway share expressed conditions by chance. Avoiding such accidental detection, we set the value  $\theta_S$  as 10. The probability of the existence of the subgraph satisfying  $\theta_S = 10$  and  $\theta_I = 3$  in a random subgraph is very low.

2) *Results of cooperative pathways*: Table I presents details of the enumerated CPs. Because our algorithm is for enumeration and not ranking, the table is not arranged in any specific order. The YO1 to YO6 networks are cooperative pathways extracted from overexpressed conditions, and YS1 and YS2 are from suppressed conditions. For example, YO1 contains three disconnected subgraphs consisting of 48 edges and 35 genes. All the genes in YO1 are highly expressed in three conditions: stationary phase 6 hours (25 degree), 12 hours (25 degree) and 2 days (30 degree). Although we have no assumption about the choice of conditions in our algorithm, common conditions of each subgraph in Table I are mutually similar. A column GO shows the

Table I  
OVEREXPRESSED AND SUPPRESSED COOPERATIVE PATHWAYS FROM YEAST STRESS DATASET.

ID	# of subgraphs	# of edges	# of genes	common pattern conditions	GO (Biological process)	<i>p</i> -value
YO1	3	48	35	Stationary phase 6h (25 degree), 12h (25 degree), 2d (30 degree)	Disaccharide	3.2e-10
YO2	3	44	31	Stationary phase 10h (30 degree), 12h (30 degree), 3d (30 degree)	Cellular carbohydrate biosynthetic process	6.8e-10
YO3	3	43	32	Heat shock 17 to 37 degree, Stationary phase 10h (30 degree), 12h (30 degree)	Acetyl-CoA metabolic process	6.7e-16
YO4	3	42	31	Stationary phase 5d (30 degree), 5d (25 degree), 7d (25 degree)	Amino catabolic process	1.3e-9
YO5	2	55	39	Heat shock 17 to 37 degree, 21 to 37 degree, 25 to 37 degree	Disaccharide metabolic process	1.1e-15
YO6	2	48	26	Nitrogen depletion 1h, 2h, 4h	Glutamine family amino acid metabolic process	1.1e-15
YS1	3	88	57	Stationary phase 5d (30 degree), 13d (25 degree), 22d (25 degree)	Glycolysis	6.1e-18
YS2	2	21	18	Stationary phase 6h (30 degC), 2d (30 degC), 3d (30 degC)	Ribonucleotide metabolic process	2.0e-12

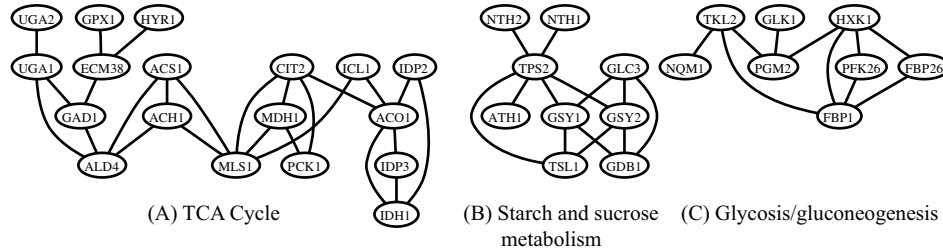


Figure 4. Genes and their relations in YO1 network in Table I. All genes are highly expressed under stationary phases. These pathways are cooperative under the stationary phases.

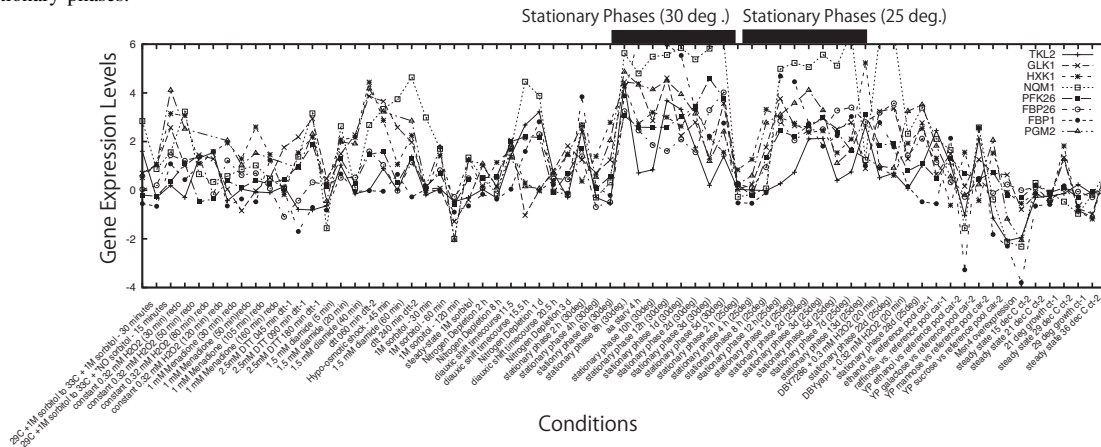


Figure 5. Gene expressions of genes in Figure 4(C) under various stress environments. All genes are highly expressed under stationary phases.

most associated biological process term in Gene Ontology (GO) [15] with the CP. YO1 is associated with disaccharide metabolic process, and its *p*-value is 3.2e-10 (binomial test). Because all the CPs we extracted are highly associated with at least one GO term, our CPs are implicated as biologically meaningful networks.

Details of YO1 are described in Figures 4 and 5. The YO1 consists of three disconnected subgraphs; and Figure 4 depicts the three subgraphs. In this figure, vertices are genes, and edges are pathway relations between the genes. The three subgraphs are separated, but all the genes are highly expressed under stationary phases. Gene expressions of the genes in Figure 4(C) are described at Figure 5 in which 39 conditions are selected randomly from 173 available conditions. Lines indicate genes, and x-axis and y-axis respectively show observed conditions and gene expression

levels. We can confirm that most genes shown in Figure 5 are highly activated under stationary phases.

We check the associations between extracted subgraphs and the KEGG pathway. In Figure 4, Figures 4(A) and (B) are respectively associated with the TCA cycle and starch and sucrose metabolism. Genes in Figure 4(C) are associated with multiple pathways. Because we combined all the pathways using gene associated with multiple pathways, the subgraph across multiple pathways can be extracted. By checking the association between the subgraph and the Gene Ontology annotations, 7 out of 8 genes in the subgraph are members of the glucose metabolic process (The *p*-value of this distribution is 1.03e-12 in the binomial test). Therefore, the network might indicate a new pathway involved in the glucose metabolic process.

All the common conditions of YO5 is associated with

heat shock stresses. This network consists of two separate subgraphs: the starch and sucrose metabolism pathway and the pyruvate metabolism pathway. The starch and sucrose metabolism pathway is also activated in the stationary phase. The results showed that this pathway works in association with the TCA cycle at the stationary phase. However, under heat shock stress, the starch and sucrose metabolism cooperates with pyruvate metabolism, which is an element involved in activating the TCA cycle, and generate energy to accelerate various metabolic processes. According to change of temperature caused by heat shock, energy would be generated increasingly to initiate metabolic cycles within a cell. These results show good harmony with currently available biological knowledge.

Our method also extract networks in which genes are suppressed. YS1 and YS2 depicts the networks. Genes in YS1 are associated with the purine metabolism, glycolysis/gluconeogenesis and the glycine, serine and threonine metabolism. The glycolysis/gluconeogenesis pathway is associated with both overexpressed genes and suppressed genes under stationary phases. For example, enzyme 5.4.2.2 consists of two genes— PGM1 and PGM2— and PGM1 is suppressed under stationary phase. However, PGM2 is highly expressed. Similarly, enzyme 2.7.1.1 contains both overexpressed genes and suppressed genes. These expression changes will alter enzyme activities. Therefore, changes in enzyme usage might be detected.

Finally, no CP contains both over-expressed and suppressed conditions.

3) *Comparison with MATISSE*: We compare our result with that of MATISSE [8]. We set the largest size cluster of MATISSE as 70 because our largest CPs contains 58 genes. MATISSE extract 29 modules, among which the average number of genes is 20.6. The lowest  $p$ -value for GO biological process terms is  $8.8e-44$  (biopolymer glycosylation), which is lower than CRAPE results. The worst (highest)  $p$ -value is  $5.6e-4$ , which is higher than any CRAPE results. Therefore, CPs extracted by CRAPE are competitive with MATISSE modules.

Three subnetworks of YO1 shown in Figure 4 are associated, respectively, with three different modules in MATISSE result. Based on the MATISSE result, it is difficult to find the association that these modules are activated under the same stationary phase conditions.

#### IV. CONCLUSION

Detecting the presence of side effects is an important problems in drug discovery and biological experimental design. In this paper, we proposed the prediction of side effects by identifying “cooperative pathways” that are disconnected pathways activated under the same sets of conditions. However, it is difficult to enumerate the combination of pathways because the number of subnetworks increases exponentially according to the pathway network size. To detect them efficiently, we introduced a novel algorithm

called CREPE, which identifies large disconnected subgraphs that are activated together under common conditions. By applying CREPE to the yeast stress gene expressions and the KEGG pathway data, we discovered that the starch and sucrose metabolism pathway cooperates with the pyruvate metabolism under heat shock stresses, although it cooperates with the TCA cycle under the stationary phases.

#### REFERENCES

- [1] A. Inokuchi, T. Washio, and H. Motoda, “An apriori-based algorithm for mining frequent substructures from graph data,” in *PKDD '00*, 2000.
- [2] M. Kuramochi and G. Karypis, “Frequent subgraph discovery,” in *ICDM*, 2001, pp. 313–320.
- [3] X. Yan and J. Han, “gspan: Graph-based substructure pattern mining,” in *ICDM '02*, 2002, p. 721.
- [4] R. Agrawal and R. Srikant, “Fast algorithms for mining association rules,” in *Proc. 20th Int. Conf. Very Large Data Bases, VLDB*, 1994, pp. 487–499.
- [5] J. Han, J. Pei, and Y. Yin, “Mining frequent patterns without candidate generation,” in *SIGMOD '00*, 2000, pp. 1–12.
- [6] S. Basu, M. Bilenko, and R. J. Mooney, “A probabilistic framework for semi-supervised clustering,” in *KDD '04*, 2004, pp. 59–68.
- [7] M. Shiga, I. Takigawa, and H. Mamitsuka, “A spectral clustering approach to optimally combining numerical vectors with a modular network,” in *KDD '07*, 2007, pp. 647–656.
- [8] I. Ulitsky and R. Shamir, “Identification of functional modules using network topology and high throughput data,” *BMC Systems Biology*, vol. 1, 2007.
- [9] F. Moser, R. Colak, A. Rafiey, and M. Ester, “Mining cohesive patterns from graphs with feature vectors,” in *SDM '09*, 2009.
- [10] K. Hashimoto, I. Takigawa, M. Shiga, M. Kanehisa, and H. Mamitsuka, “Incorporating gene functions as priors in model-based clustering of microarray gene expression data,” *Bioinformatics*, vol. 24, no. 16, pp. i167–i173, 2008.
- [11] M. Seki and J. Sese, “Identification of active biological networks and common expression conditions,” in *BIBE '08*, 2008.
- [12] A. Subramanian, P. Tamayo *et al.*, “Gene set enrichment analysis: A knowledge-based approach for interpreting genome-wide expression profiles,” *Proc. Natl. Acad. Sci.*, vol. 102, pp. 15 545–15 550, 2005.
- [13] M. Kanehisa and S. Goto, “KEGG: Kyoto Encyclopedia of Genes and Genomes,” *Nucleic Acids Research*, vol. 28, no. 1, pp. 27–30, 2000.
- [14] A. P. Gasch *et al.*, “Genomic expression programs in the response of yeast cells to environmental changes,” *Mol. Biol. Cell*, vol. 11, no. 12, pp. 4241–4257, 2000.
- [15] M. Ashburner, C. A. Ball, J. A. Blake, D. Botstein, H. Butler *et al.*, “Gene ontology: tool for the unification of biology. the gene ontology consortium.” *Nat Genet*, vol. 25, no. 1, pp. 25–29, May 2000.