빈발 유전자 발현 패턴과 연쇄 규칙을 이용한 유전자 조절 네트워크 구축

이 헌 규[†] · 류 근 호[††] · 정 두 영[†††]

요 약

유전자들의 그룹은 복잡한 상호작용들을 통해 세포의 기능이 조절되며 이러한 상호작용을 하는 유전자 그룹들을 유전자 조절 네트워크(GRNs: Gene Regulatory Networks)라고 한다. 이전의 유전자 발현 분석 기법인 군집화와 분류는 단지 상동성에 의한 유전자들 사이의 소속을 결정하는 데에는 유용하나 분자 활동에서의 같은 클래스에서 발견되어지는 유전자들 사이의 조절 관계를 식별할 수 없다. 더욱이 유전자들이 어떻게 연관되는지와 유전자들이 서로 어떻게 조절하는지에 대한 매커니즘의 이해가 필요하다. 따라서 이 논문에서는 시계열 마이크로어레이 데이터로부터의 유전자들의 조절 관계를 발견하기 위해서 빈발 패턴 마이닝과 연쇄 규칙을 이용한 새로운 접근법을 제안하였다. 이 기법에서는 먼저, 빈발 패턴 마이닝 적용을 위한 적절한 데이터 변환 방법을 제안하였고 FP-growth을 이용하여 유전자 발현 패턴들을 발견한다. 그런 다음, 연쇄 규칙을 이용하여 빈발한 유전자 패턴들로부터 유전자 조절 네트워크를 구축하였다. 마지막으로 제안된 기법의 검증은 공개된 유전자들의 조절 관계와 실험 결과의 일치함을 보임으로써 평가하였다.

키워드 : 유전자 조절 네트워크, 빈발 패턴 마이닝, 연쇄 규칙, 유전자 상호작용

Constructing Gene Regulatory Networks using Frequent Gene Expression Pattern and Chain Rules

Heon Gyu Lee[†] · Keun Ho Ryu[††] · Doo Young Joung[†††]

ABSTRACT

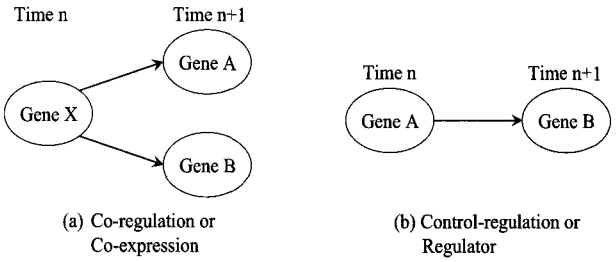
Groups of genes control the functioning of a cell by complex interactions. Such interactions of gene groups are called Gene Regulatory Networks(GRNs). Two previous data mining approaches, clustering and classification, have been used to analyze gene expression data. Though these mining tools are useful for determining membership of genes by homology, they don't identify the regulatory relationships among genes found in the same class of molecular actions. Furthermore, we need to understand the mechanism of how genes relate and how they regulate one another. In order to detect regulatory relationships among genes from time-series Microarray data, we propose a novel approach using frequent pattern mining and chain rules. In this approach, we propose a method for transforming gene expression data to make suitable for frequent pattern mining, and gene expression patterns are detected by applying the FP-growth algorithm. Next, we construct a gene regulatory network from frequent gene patterns using chain rules. Finally, we validate our proposed method through our experimental results, which are consistent with published results.

Key Words : Gene Regulatory Network, Frequent Pattern Mining, Chain Rules, Gene Interaction

1. 서 론

최근 마이크로어레이 기술을 이용하여 유전자의 발현 데이터로부터 유전자의 상호작용 조절 네트워크를 추론 하는 기법이 세포활동의 기작을 밝혀내는 데에 사용되고 있다. 유전자들의 상호 조절 작용은 세포 안에서 단백질과 간접적으로 다른 유전자들에 의해 조절된다. 유전자들의 그룹은 복잡한 상호작용들을 통해 세포의 기능이 조절되며 이러한 상호작용을 하는 유전자 그룹들을 유전자 조절 네트워크(GRNs: Gene Regulatory Networks)라고 한다. 실제로 유전자 발현은 연속적인 생물학적 프로세스이고 종종 up-regulation과 down-regulation으로 추상화된다. 유전자 상호작용들은 두 가지의 중요한 그룹으로 나눌 수 있다. 첫째로 Co-regulation이란 (그림 1)(a)과 같이, 두 가지의 유전자들의 발현(Gene A, B)이 또 다른 유전자(Gene X)에 의해 조절되는 것을 의미한다. 둘째, Control regulation(regulator)이란 하나의 유전자(Gene A)는 전사 레벨에서 다른 유전자

※ 이 논문은 2007년도 교육인적자원부 지방연구중심대학 육성사업의 지원에 의하여 연구됨.
† 준 회 원 : 충북대학교 대학원 전자계산학과 박사과정
†† 종신회원 : 충북대학교 전기전자 컴퓨터공학부 교수
††† 정 회 원 : 충북대학교 전기전자 컴퓨터공학부 교수
논문접수 : 2006년 8월 2일, 심사완료 : 2007년 1월 5일



(그림 1) 유전자의 발현 패턴

(Gene B)의 발현을 조절하는 것을 의미한다((그림 1)(b))[1, 2]. 추가적으로 유전자는 하나 이상의 activator와 inhibitor라고 하는 조절자로 구분된다. Activator는 유전자의 발현을 위한 신호로서 activator가 없으면 낮은 발현 상태만을 나타내며, 반대로 inhibitor는 유전자의 발현을 억제하는 기능을 한다. (그림 1)(b)처럼 유전자 A가 발현한 뒤에 유전자 B가 발현하게 될 경우, 유전자 A를 유전자 B를 조절하는 activator라고 한다. 유전자 A가 발현한 뒤에 유전자 B가 발현되지 않은 상태로 변하게 될 경우에 유전자 A를 inhibitor라고 한다[3].

이러한 유전자 상호작용 조절 네트워크 구축에 대한 기존 연구로는 첫째, 시계열 접근 방법을 들 수 있다. 이 접근 방법은 어느 특정 시점에 있어서 유전자의 발현 패턴이 그 이전 시점에서 모든 다른 유전자 발현 패턴의 함수로서 모델화 하는 방법이다. 그러나 이 방법은 마이크로어레이 데이터의 특징인 적은 time-point와 많은 수의 유전자가 문제시되며, 계산하기 복잡한 차원의 문제로 제시된다. 따라서 [4]에서는 발현에서 유의한 차이를 보이지 않는 유전자들을 제외시켜 차원을 줄이는 선형 모델법을 제안하였으며, 특이치 분해(SVD: Singular Value Decomposition)를 이용하여 차원의 문제를 해결한 [5]는 상호작용을 알아내기 위해 SVD를 실시하여 극히 적은 수의 유전자만을 남기고, 이러한 유전자만으로 상호작용 행렬을 풀어 유전자의 상호작용을 쉽게 발견한다. 둘째, 기계학습을 이용한 새로운 방법으로 베이지안 네트워크가 있다[1, 5, 6]. 그 중 [1]에서 제안한 방법은 희귀(sparse) 후보 기법과 모델평균화 기법을 이용한 통계적 접근법이다. 유전자를 그의 발현이 증가하는 것과 감소하는 것으로 단순화하여 확률적 네트워크를 추정하고 다른 모델에 같은 데이터를 적용할 때에도 공통적으로 얻어지는 결과를 통해 탐색하였다. 또한 유전자의 상태 전이가 동기화 되었다는 것과 유전자의 활동이 단지 두 가지라는 가정 하에서 유전자 조절 네트워크를 부울 네트워크(boolean network)의 형태로 추론하는 방법이 [7]에서 제안하였다. 여기서는 유전자의 활동 레벨을 두 가지 상태(on/off)로 놓고 어떠한 유전자의 조합이 한 유전자의 다음 단계 활동 수준을 결정하는가를 알아내기 위해 상호정보량(mutual information)을 활용한 적용하였으며 그러한 조합을 알아내어 유전자 조절 네트워크를 구성하였다. 그러나 베이지안 네트워크를 이용한 조절 네트워크 추론은 베이지안의 이론적 근거와 통계적 안정성을 가지지만, 마이크로어레이 데이터의 많은 양의 유전자들을 추론함에 있어 충분한 양의 훈련데이터를 얻는 것이 어려우며, 이로 인해 찾아낸 네트워크의 관계 중 높은 양성 오류(false positive)율의 잘못된 예측을 하게 된다.

이 논문에서는 시계열 마이크로어레이 데이터로부터의 조절 네트워크 구축을 위해 빈발패턴 마이닝과 연쇄 규칙을 이용한 기법을 제안한다. 이전의 마이크로어레이 분석 기법인 클러스터링과 분류기법[8-11]에서는 단지 발현 패턴이 유사한 유전자들의 그룹핑과 기능이 알려진 유전자들로부터 모델을 학습시키고 이러한 모델로부터 새로운 유전자들의 기능을 예측하는 것으로 한정된다. 따라서 이러한 방법을 통해서 유전자들이 어떻게 연관되는지와 유전자들 간의 어떤 조절 관계를 갖는지에 대한 매커니즘을 이해할 수 없다. 따라서 마이크로어레이 데이터에서 빈발한 유전자들의 패턴을 발견하고 이러한 패턴들을 이용하여 연속적인 조건부 확률을 적용한 통계적 기법인 연쇄 규칙을 유도함으로써 유전자들 간의 상호 조절 관계를 발견할 수 있다. 또한 제안된 기법은 알려지지 않은 유전자들의 조절 관계를 표현할 수 있고 이로부터 아직 알지 못하는 생물학적 정보를 얻을 수 있다. 제안된 조절 네트워크 구축 방법을 위해 논문은 다음과 같은 내용으로 구성된다.

- 연속적인 실수 값의 유전자 발현 데이터에서 빈발 패턴 탐사가 가능하도록 하기 위해서 데이터 변환 기법인 이산화 방법을 제안한다.
- 전처리된 유전자 발현 데이터로부터 빈발한 패턴 탐사를 위해 현재까지 성능이 가장 우수한 FP-growth 기법을 적용한다.
- 탐사된 유전자 패턴은 대용량이며 많은 중복 패턴들을 포함한다. 따라서, 유용한 패턴만을 추출하기 위해 패턴 응집도(*PC*) 측정 지표를 제안하며 압축 패턴 트리를 이용, 중복 패턴들을 제거한다.
- 조건부 확률과 결합 확률을 이용한 연쇄 규칙을 이용하여 빈발한 유전자 패턴들로부터 유전자 조절 네트워크 구축 방법을 제안한다.
- 제안된 기법의 검증은 공개된 유전자들의 조절 관계와 실험 결과의 일치함을 보임으로써 평가되고, 그 평가 기준으로써 Precision, Recall, F-Measure, MAE를 사용한다.

논문의 효과적인 이해를 위해서 논문의 구성은 다음과 같이 구성하였다. 2장에서는 빈발 패턴 마이닝 적용을 위한 전처리 단계로 유전자 발현 데이터의 이산화 방법을 기술하고 3장에서는 FP-growth 기법을 적용한 유전자의 빈발 패턴 탐사과정의 알고리즘들을 설명한다. 4장에서는 생성된 유전자 발현 패턴들로부터 연쇄 규칙 적용 알고리즘과 조절 네트워크 구축 방법을 기술한다. 제안한 유전자 조절 네트워크 구축 기법은 Yeast 데이터에 적용하여 실험한 후 그 결과의 분석은 5장에 기술한다. 마지막으로 6장에서는 이

논문에 대한 전체적인 결론을 맺는다.

2. 유전자 발현 마이크로어레이 데이터 전처리

이 장에서는 유전자 발현 마이크로어레이의 데이터에서 연속적인 발현 값에 대한 이산화 방법, 그리고 빈발 패턴 탐사가 가능하도록 유전자 발현 배열을 트랜잭션 데이터베이스화 하는 과정을 기술한다.

마이크로어레이 데이터 D는 (그림 2)와 같이<*gene, time-point*> 형태의 $n \times m$ 데이터 행렬로 표현되어지며 n개의 유전자(또는 *probe*), m개의 실험 샘플로 구성된다.

일반적으로 유전자에 대한 발현 량에 대한 값은 실수 값이며 시계열 유전자 데이터의 경우, m개의 샘플은 서로 다른 시점에서의 측정된 연속적인 수치이다.

빈발 패턴 마이닝을 위한 일반적인 장바구니 데이터와 유전자들의 발현 패턴에 대한 마이닝 문제는 중요한 차이점을 갖는다. 첫째로, 장바구니 분석에서는 상당히 많은 수의 트랜잭션을 포함하며 상대적으로 적은 수의 항목들을 포함한다. 또한 특정 트랜잭션에서의 항목집합은 이진 값(0, 1)으로 표현되며, 항목들의 구매 여부만을 표현한다. 반면에 유전자 발현 데이터는 많은 수의 항목(genes or ORFs)[1]들을 포함한다. 그러나 장바구니 분석에서의 트랜잭션에 해당하는 실험의 측정치들의 수는 훨씬 적은 수이다. 또한 장바구니 데이터에서는 모든 트랜잭션에 전체 항목들을 포함하지 않지만, 유전자 발현 데이터의 실험 측정치에서는 반드시 모든 유전자들이 실수 값으로 표현되어 있다. 마이크로어레이의 데이터의 샘플은 다음 (그림 3)과 같으며, 유전자를 의미하는 ORF (Open Reading Frame)은 항목으로, 각 time point는 트랜잭션으로 간주한다.

(그림 1)의 유전자 발현 데이터를 이산화하면 특정 기준치에 따라 이진 값으로 변환하게 된다. 이진 값으로 변환된 데이터는 이진수 배열로 만들게 되고, 데이터의 표현이 간

$$D = \begin{bmatrix} D_{1,1} & D_{1,2} & \cdots & D_{1,m} \\ D_{2,1} & D_{2,2} & \cdots & D_{2,m} \\ \cdots & \cdots & \cdots & \cdots \\ D_{n,1} & D_{n,2} & \cdots & D_{n,m} \end{bmatrix}$$

(그림 2) 마이크로어레이 데이터의 행렬 표현

ORF	Time 1	Time 2	Time 3	Time 4	Time 5	Time 6	Time 7
YHR007C	1.12	1.19	1.32	0.88	0.84	0.38	0.43
YBR218C	1.18	1.23	0.77	0.75	0.79	0.71	1.7
YAL051W	0.97	1.32	1.33	1.18	1.12	0.88	0.93
...	...	...	...	...	...	...	...
YAL055W	0.68	1	0.92	0.96	0.81	1.28	1.85

(그림 3) Time-series Microarray 데이터

1) Yeast 실험 데이터의 경우 6300개 이상의 유전자들을 포함한다.

	t_1	t_2	t_3	t_4
g_1	0	1	0	1
g_2	1	0	0	1
g_3	0	1	0	1
g_4	1	1	1	1

→

TID (time point)	Itemset
t_1	g_2, g_4
t_2	g_1, g_3, g_4
t_3	g_4
t_4	g_1, g_2, g_3, g_4

(그림 4) 유전자 배열로부터의 트랜잭션 생성 과정

결하게 되므로 빈발 패턴과 같은 유전자의 발현 패턴을 추출하는 것이 가능하다. 데이터 변환은 상대적인 유전자의 발현 값에 대해 3가지의 구간으로 이산화 한다. 만약, 발현 비율 값에 따라 유전자는 *gene-up, gene-down, unchange*로 3가지로 표현한다. 유전자 발현 비율에 대해 항목으로 표현이 되면, 이진수 배열로 변환하여 트랜잭션으로 나타낸다. 특정 time-point를 나타내는 트랜잭션에서 유전자의 발현 비율이 1보다 크다면, 항목의 값을 *gene-up*=1로 하고 *gene-down*와 *unchange*를 0으로 설정한다. 발현 비율이 1과 0사이의 값일 경우에는 *gene-down* 을 1로, *gene-up*과 *unchange*을 0으로 변환한다. 또한 유전자가 1의 비율과 같다면 *unchange*을 1로 나머지 *gene-up*과 *gene-down*을 0으로 설정한다. 유전자 발현 비율에 대해 항목으로 표현이 되면, 이진수 배열로 변환하여 트랜잭션으로 나타낸다. 유전자 발현 데이터의 트랜잭션으로의 변환 과정은 (그림 4)와 같다.

3. 상호작용 유전자 발견을 위한 빈발 패턴 마이닝

이 장에서는 특정 시점에서의 유사한 발현 패턴을 갖는 유전자들을 발견하기 위해서 유전자 발현 마이크로어레이 데이터에서의 빈발한 유전자 패턴 탐사 기법을 소개하며, 탐사된 패턴들 사이의 유용성 측정을 위한 새로운 측정치인 패턴 응집도를 제안한다. 또한, 생성된 대량의 패턴들에서 중복 패턴 제거 및 메모리의 효율적인 패턴 저장을 위한 압축 패턴 트리 구조를 제안한다. 먼저 유전자 발현 데이터에서의 빈발 패턴 추출을 위한 유전자 패턴에 대한 정의를 내리고 이로부터 빈발 패턴 탐사 과정 문제를 단계별로 정의한다.

[정의 1] 유전자 패턴(gene pattern): 유전자를 하나의 항목(*item*)으로 가정하고 유전자의 패턴을 나타내는 항목집합(itemset)을 $P=\{i_1, i_2, ..., i_n\}$라 할 경우, $1 \le j \le n$인 i_j은 하나의 유전자를 표현한다. ■

[정의 2] 부분 패턴(sub pattern), 상위 패턴(super pattern): 패턴, $P=\{i_1, i_2, ..., i_n\}$가 조건, $i_1 = i'_{k_1}$, $i_2 = i'_{k_2}$, $i_n = i'_{k_n}$을 만족하는 $k_1 < k_2 ... < k_n$인 정수들이 존재한다면 P는 다른 패턴 $P' = \{i'_1, i'_2, ..., i'_m\}$의 부분 패턴이라고 하며 반대로, P'을 상위 패턴이라고 한다. ■

예를 들어 $P=\{p_2, p_4, p_5, p_6\}$는 $P' = \{p_1, p_2, p_3, p_4, p_5, p_6, p_7\}$의 부분 패턴이다. (두 유전자 패턴에는 $(p_2, i_1 = i'_2), (p_4, i_2 = i'_4), (p_5, i_3 = i'_5), (p_6, i_4 = i'_6)$인 정수 $k_1 = 2 < k_2 = 4 < k_3 = 5 < k_4 = 6$

가 존재하기 때문이다.)

[정의 3] 빈발 패턴(FPs: Frequent Patterns): 빈발 패턴, *FPs*란 임계값인 최소지지도(Min_{sup})을 만족하는 각 트랜잭션의 부분 패턴(*sub pattern*)이다. ■

결론적으로 유전자 발현 데이터로부터의 빈발 패턴 탐사는 사용자가 미리 지정한 최소지지도를 만족하는 모든 빈발한 유전자들의 집합을 탐사하는 문제이다.

3.1 유전자 발현 프로파일 데이터로부터의 빈발 패턴 탐사 알고리즘

빈발 패턴(FP)의 문제 정의와 함께, 기본적인 패턴탐사 알고리즘의 구조를 묘사할 수 있다. 이 절에서는 [12]에서 소개된 FP-growth 방법을 이용하여 사용자 기반의 최소지지도를 만족하는 모든 빈발 패턴 탐사 알고리즘을 기술한다.

FP-tree는 빈발 패턴에 대한 지지도를 저장하는 *Prefix* 트리 구조이며 상위 노드들은 높은 지지도 값을 가지는 항목들이 위치하고 낮은 지지도의 노드일수록 하위 노드에 위치하는 방식으로 트리를 구성한다. FP-tree는 각 노드의 항목들의 카운트 값을 유지하기 위해서 해더 테이블 데이터

Input: (1) Expression data set D ; (2) minimum support Min_{sup}.
Output: *FP-tree* corresponding to and satisfying Min_{sup}.
1) Scan D once and collect the set of frequent items F and their supports.
2) Sort F in support descending order as L, the *list* of frequent items.
3) If several items have the same support, and their names are numbers,
sort the items in ascending order of their names.
4) Create the root R of a new **FP-tree** and label it as "null".
5) Create frequent-item header table with | F | entries. Set all head of
node-link pointers to *null*.
6) **for each** transaction data $d \in D$ **do** // Read D the second time.
7) Select only frequent items of d into a record P ;
8) Sort P in the order of L ;
9) Call **insert_tree** (P, c_d, R) ;
10) **end for**

(그림 5) FP-tree 구성 알고리즘

Procedure insert_tree (P, c, R);
1) Let $P = [p \mid P\text{-}p]$, where p is the first element of P, and P-p is the remaining list.
2) **if** R has a child N such that $N.item\_name = p$ **then**
3) $N.count = N.count + 1$;
4) **else** {
5) create a new node N;
6) $N.count = 1$; $N.item\_name = p$;
8) $N.parent = R$; $N.node\text{-}link = H(p).head$;
9) $H(p).head = N$;
10) }
11) $H(p).count = H(p).count + 1$;
12) **if** $P\text{-}p \neq \phi$ **then**
13) Call **insert_tree**($P\text{-}p$, c, N) recursively.

(그림 6) insert_tree( ) 프로시저

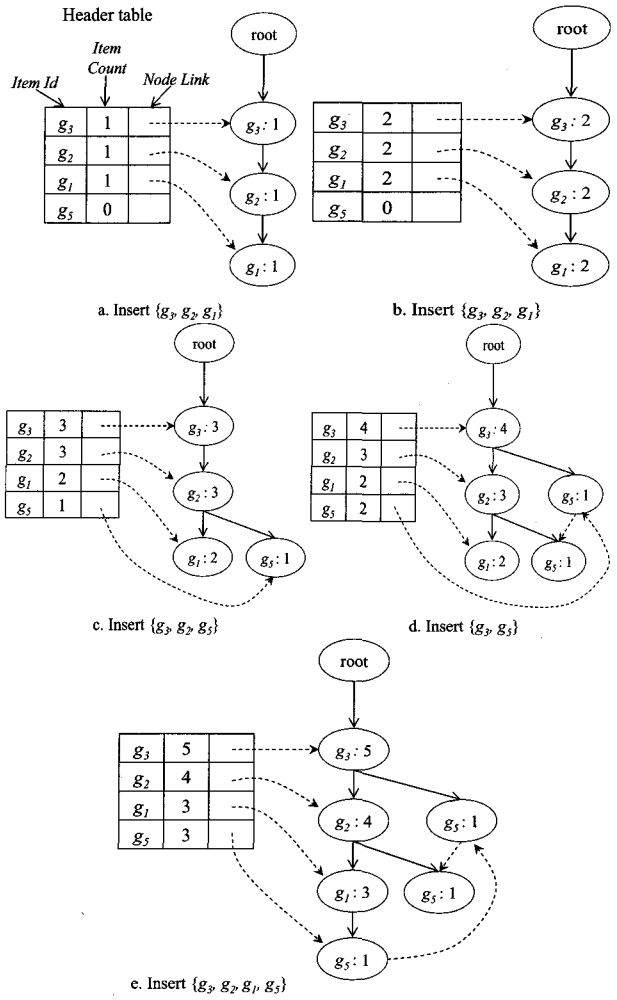
〈표 1〉 유전자 발현 프로파일 데이터에 대한 트랜잭션 DB의 예

TID	Transaction (Genes)	Inserted Patterns
1	$\{g_1, g_2, g_3, g_4\}$	$\{g_3, g_2, g_1\}$
2	$\{g_1, g_2, g_3\}$	$\{g_3, g_2, g_1\}$
3	$\{g_2, g_3, g_5\}$	$\{g_3, g_2, g_5\}$
4	$\{g_3, g_5, g_6\}$	$\{g_3, g_5\}$
5	$\{g_1, g_2, g_3, g_5\}$	$\{g_3, g_2, g_1, g_5\}$

구조를 가지며, 트리에 삽입되는 모든 패턴들은 항목들의 카운트 값을 포함한다. 유전자 발현 프로파일 데이터로부터의 모든 빈발한 패턴 탐사를 위한 Prefix 기반의 알고리즘인 FP-tree 구성 알고리즘은 (그림 5)이고 패턴의 트리 삽입 프로시저는 (그림 6)이다.

예를 들어 <표 1>이 전처리된 유전자 발현 프로파일 데이터이고 최소지지도가 2일 경우, FP-tree의 구성은 그림 7과 같다.

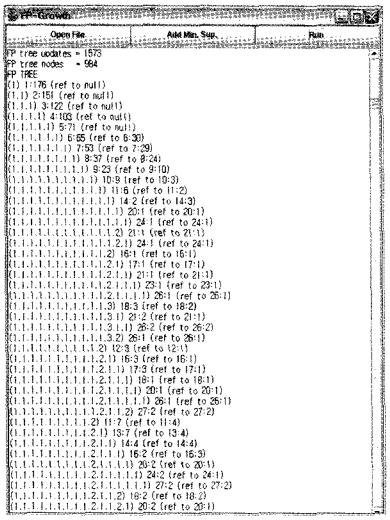
<표 1>의 *Inserted Patterns*는 트리에 삽입될 패턴들의 순서를 L의 기준에 따라 나타낸다. TID가 1인 트랜잭션 $\{g_1, g_2, g_3, g_4\}$은 지지도를 만족 못하는 g_4을 제외하고 (그림



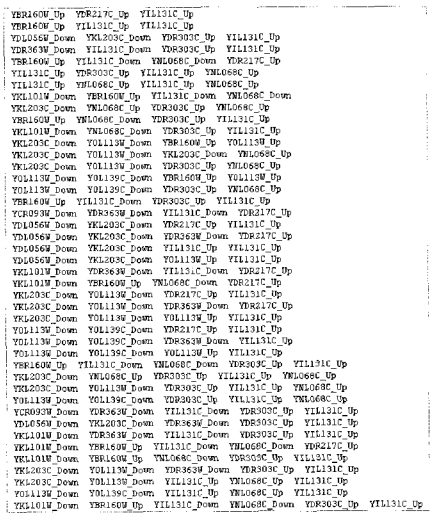
(그림 7) FP-tree의 구성과정

〈표 2〉 FP-tree로부터의 빈발 패턴 마이닝

항목	조건부 패턴 베이스	조건적 FP-tree	빈발 패턴
g_5	$\{(g_3,g_2,g_1:1),(g_3:1),(g_3,g_2:1)\}$	$<g_3:3,g_2:2>$	$\{g_5:3\},\{g_3,g_5:3\},\{g_2,g_5:2\},\{g_3,g_2,g_5:2\}$
g_1	$\{(g_3,g_2:3)\}$	$<g_3:3,g_2:3>$	$\{g_1:3\},\{g_3,g_1:3\},\{g_2,g_1:3\},\{g_3,g_2,g_1:3\}$
g_2	$\{(g_3:4)\}$	$<g_3:4>$	$\{g_2:4\},\{g_3,g_2:4\}$
g_3	$\varnothing$	$\varnothing$	$\{g_3:5\}$



(a) 빈발 패턴 탐사 과정



(b) 발견된 빈발 패턴의 예

(그림 8) FP-growth를 이용한 유전자 패턴 탐사 과정

7)(a)와 같이 트리에 삽입되며 해더 테이블의 카운트를 1로 할당한다. 두 번째 삽입될 패턴은 이전의 패턴과 동일하므로 단순히 카운트 값만을 1씩 증가시킨다. 세 번째 패턴 $\{g_3,g_2,g_5\}$은 이미 삽입된 $\{g_3,g_2,g_1\}$ 패턴의 공통의 접두부 $\{g_3,g_2\}$을 공유하므로 두 접두부에 대한 카운트는 1씩 증가시키고 $\{g_5\}$에 대한 새로운 가지를 생성하여 카운트로 1로 할당한다. TID 4, 5에 대해서도 같은 방식으로 트리에 삽입되며 최종 구성된 트리는 (그림 7)(e)이다.

[12]에서 제안된 *FP-tree*를 이용한 빈발 패턴 마이닝의 전 과정을 <표 2>에 나타내었고 그 과정의 예는 (그림 8)이다.

3.2 중복 패턴 제거 및 패턴 저장을 위한 CP-tree 알고리즘

모든 빈발 패턴이 알고리즘에 의해 생성된 후, 각 패턴의 유용성 측정을 위해서 지지도 외에 새로운 측정치인 패턴 응집도(*PC: Pattern Cohesion*)를 정의하며, 이는 압축 패턴 트리 생성 시의 중복패턴 및 불필요한 패턴의 제거에 사용된다.

[정의 4] 패턴 응집도(*PC*: Pattern Cohesion): 길이가 n인 패턴($p=p_1,\ldots,p_n$)에 대해, 패턴 p의 PC는 식 (1)과 같이 정의되며, 모든 빈발 패턴들은 PC에 대해 우선순위가 결정된다.

$$PC(p_1,\ldots,p_n)=\frac{Count(p_1,\ldots,p_n)}{\sqrt[n-1]{Count(p_1)\times\ldots\times Count(p_n)}} \quad \text{식(1)}$$

■

식(1)은 문서 분류(*text classification*) [13]에서 두 단어(*two-word*) 사이의 상관성 측정치인 식(2)을 n개의 길이를 갖는 항목집합의 문제로 확장한 것이다.

$$Cohesion(w_i,w_j)=\frac{P(w_i,w_j)}{\sqrt{P(w_i)\times P(w_j)}} \quad \text{식(2)}$$

모든 생성된 패턴들은 대용량의 패턴 리스트이며, 많은 중복된 패턴들을 포함한다. 따라서 효율적인 패턴들의 저장과 PC에 기반한 중복 패턴 제거를 위해, [14]에서 소개된 트리를 변형한 압축 패턴 트리, *CP-tree*(*Compressed Pattern tree*) 데이터 저장 구조를 제안한다. *CP-tree* 구조는 압축된 패턴 저장이 가능하고 패턴들 사이의 부분 패턴(*sub sequence pattern*)/상위 시퀀스 패턴(*super sequence pattern*) 관계들을 반영한다.

*CP-tree*의 구조는 *FP-tree* 구조와 유사하나 다음의 두 가지 다른 특징을 갖는다.

- 모든 패턴에 대해, *CP-tree*는 지지도뿐만 아니라 각 패턴의 응집도인 PC를 포함한다.
- 단지 마지막 리프 노트만이 모든 패턴들의 속성 정보만을 가지며, 하나의 패턴은 리프로부터 루트까지의 패스이다.

모든 패턴들이 *CP-tree*에 삽입될 때, 동시에 중복 패턴들의 삭제가 일어나며, *CP-tree* 구성 알고리즘은 (그림 9)와 같다.

Input: a set FP of frequent patterns.
Output: a subset $FP_p \in FP$ containing non-redundant patterns
1) create a root of CP-tree;
2) **for** each pattern $p \in FP$ **do**
3) **if** there is no sub pattern p_{sub} of p s.t. $PC(p_{sub}) > PC(p)$ **then**
4) if there are super patterns p_{supp} of p s.t. $PC(p_{supp}) < PC(p)$ **then**
5) delete all p_{supp};
6) insert p into tree;
7) **end for**
8) Traverse tree and insert all patterns into FP_p.

(그림 9) 압축패턴 트리 알고리즘

〈표 3〉 빈발 패턴과 PC 예

$p-id$	*Frequent Pattern*	*PC Measure*
1	p_1, p_2, p_3, p_4, p_7	0.70
2	p_2, p_3, p_4, p_8	0.60
3	p_2, p_3, p_4, p_5, p_8	0.40
4	p_1, p_2, p_6, p_7	0.60
5	p_2, p_3, p_8	0.65
6	p_2, p_3, p_4, p_7	0.80

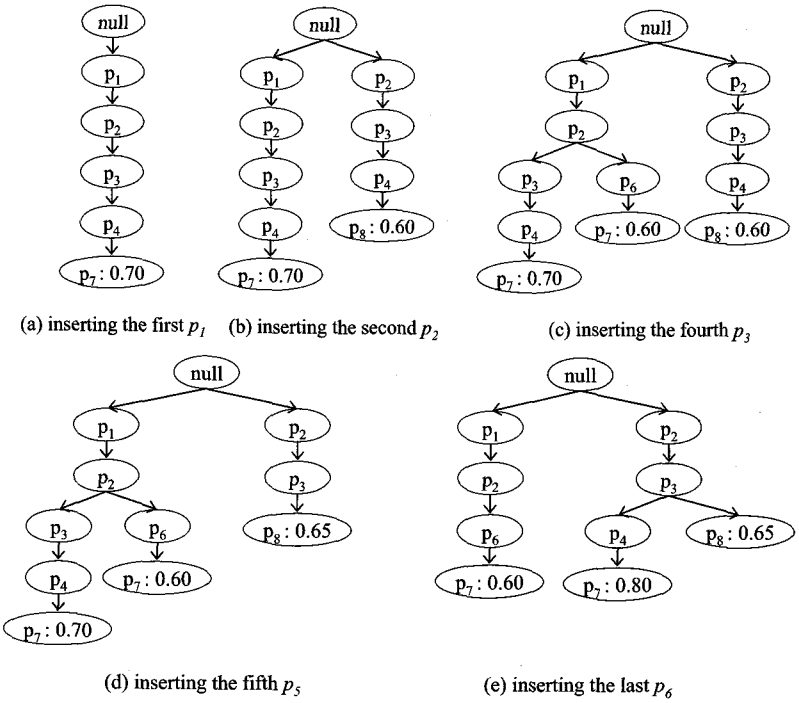
예를 들어 (그림 8)의 알고리즘 수행 결과 모든 빈발 패턴이 <표 3>과 같다고 할 경우, *CP-tree*의 구성과 중복 패턴의 제거 과정은 다음과 같다.

p-id 1인 첫 번째 $p_1 = \{p_1, p_2, p_3, p_4, p_7\}$가 처음으로 트리에 삽입되고(그림 10)(a) 다음의 두 번째 p_2가 트리에 삽입된다. p_2는 p_1보다 작은 *PC*값을 가지나 p_1의 부분 패턴(*sub sequence pattern*)이므로 제거되지 않는다(그림 10)(b). 세 번째 $p_3 = \{p_2, p_3, p_4, p_5, p_8\}$는 p_2의 *PC*값이 낮은 상위(*super sequence pattern*)이다. 따라서 p_3는 중복 패턴이 되므로 트리에 삽입되지 못하고 제거된다. p_4는 어떠한 패턴(p_i)에 의해 제거되지 않으므로 첫 번째 패턴과 접두부(prefix)를 공유하며 트리에 삽입된다(그림 10)(c). (그림 10)(d)는 p_5가 삽입될 때, 이미 트리에 삽입되어 있는 p_2을 제거하며 p_5가 트리를 재구성하는 과정을 나타낸다. (p_5는 p_2보다 높은 *PC*값을 가지며, 동시에 p_2의 부분 시퀀스 패턴이므로 삽입된 패턴, p_2을 제거한다.) 마지막 패턴 p_6이 트리에 삽입되며 첫 번째 p_1는 트리에서 제거된다(그림 10)(e). 최종적으로, 남아있는 가지만이 탐사된 패턴들로 되며 3가지만을 가진다.

4. 연쇄 규칙을 이용한 유전자 상호작용 네트워크 구축

모든 패턴의 집합 *FP*와 연쇄 규칙을 이용하여 네트워크 구조를 표현한다. 이때 선택된 패턴들은 구축하고자 하는 유전자 집합의 부분 패턴들이 된다. 빈발 패턴들을 이용한 유전자 네트워크의 구성은 연쇄 규칙의 확률적 네트워크 모델로의 문제가 되며 이 장에서는 이와 관련된 확률들의 정의와 함께 *FP* 패턴 집합으로 부터의 연쇄 규칙을 적용한 근사 곱 문제를 기술한다.

[정의 6] 조건부확률(conditional probability), 결합확률(joint probability): X와 Y 두 개의 사건에 대해, 사건 Y가 일어날 확률이 이미 알려져 있을 경우에 사건 X가 일어날 확률을



(그림 10) 패턴 집합 P로부터의 CP-tree 구성 과정

조건부 확률이라고 하며, 식(3)과 같다.

$$P(X|Y)=\frac{P(X\cap Y)}{P(Y)} \quad \text{식(3)}$$

식(3)은 $P(X)P(Y|X)=P(X\cap Y)$가 성립하고, $P(Y)P(X|Y)=P(X\cap Y)$도 성립한다. X와 Y가 동시에 발생하는 확률을 결합 확률이라고 하며 조건부 확률의 수식으로 유도 될 수 있는데 이를 곱셈 법칙이라고 한다. 만약 X와 Y가 서로 독립이라고 한다면 $P(X|Y)=P(X)$이 되므로 곱셈 법칙에 대입하면 $P(Y)P(X)=P(X\cap Y)$가 성립한다. ■

[정의 7] 연쇄 규칙(chain rule): 각 사건 $A_1, A_2, ..., A_n$이 일어날 확률은 조건부 확률과 결합확률을 이용하여 식(4)와 같이 연쇄적인 조건부 확률의 곱의 표현식으로 표현할 수 있으며, 이를 연쇄 규칙(chain rule)이라고 한다.

$$P(A_1, A_2, ..., A_n)= P(A_1|A_2, A_3, ...A_n) \cdot P(A_2|A_3, A_4, ..., A_n) \cdot ... \cdot P(A_{n-1}|A_n) \cdot P(A_n) \quad \text{식(4)}$$

■

[정의 6]과 [정의 7]에 의해서 유전자 데이터로부터 네트워크 구축을 확률적 모델로 추정할 수 있다. 먼저 네트워크 구성을 위한 집합 유전자들의 집합 $G=\{g_1, g_2, ..., g_n\}$이 주어진다면, 확률 $P(G)=P(g_1, g_2, ..., g_n)$을 최대로 하는 유전자 패턴들의 곱으로서 연쇄 규칙을 적용하여 표현되어진다. 여기서의 유전자 패턴은 3.2절에서 중복패턴이 제거된 높은 질의 유전자 집합 FP의 빈발 패턴들이다.

[정의 8] 근사 곱(product approximation): [정의 7]의 연쇄 규칙을 이용하여 유전자들의 확률 $P(g_1, g_2, ..., g_n)$은 서로 다른 근사(approximation)들로 추정될 수 있다[15]. 각 근사들은 속성들에 대해 서로 다른 조건 독립 가정을 나타낸다. $P(g_1, g_2, g_3, g_4)$은 확률의 곱 $P(g_1, g_2) \cdot P(g_3, g_4|g_1)$ 또는 $P(g_1, g_2) \cdot P(g_4|g_2) \cdot P(g_3|g_1, g_4)$으로 계산되며, 이 때 두 확률을 근사 곱(product approximation)이라고 한다. ■

여기서, 특정 근사들의 선택은 빈발 패턴의 선택과 동등하다. 예를 들어 확률의 곱이 $P(g_1, g_2) \cdot P(g_3, g_4|g_1)$일 경우의 근사는 생성된 패턴 $\{g_1, g_2\}$, $\{g_1, g_3, g_4\}$을 포함한다.

4.1 연쇄 규칙을 이용한 근사 곱의 구성 알고리즘

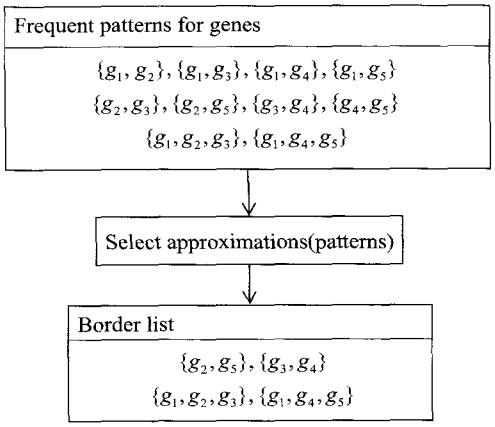
유전자 조절 상호작용 네트워크 구성을 위한 유전자들의 집합이 주어지면, 그 집합에 포함되는 모든 빈발 패턴 리스트들이 형성되며, [정의 8]의 근사 곱으로써 그 유전자 네트워크 모델의 확률을 최대로 하는 근사들을 선택할 수 있다.

[정의 9] 빈발 패턴들의 경계(border), B: B는 조절 네트워크 구성을 위한 유전자들의 집합, $G=\{g_1, g_2, ..., g_n\}$의 부분 패턴들로 빈발 패턴 집합 안에 존재하며, 그 관계는 식(5)과 같다.

$$B=\{p\in FP | p\subset G\} \quad \text{식(5)}$$

■

예를 들어 5가지의 $G=\{g_1, g_2, g_3, g_4, g_5\}$ 유전자 집합에 대한 근사 구성을 할 경우, 빈발한 모든 패턴들의 집합으로부터 G에 대한 근사들의 집합인 경계(B) 구성은 그림 11과 같다.



(그림 11) 경계(B) 리스트의 구성 예

조절 네트워크 구성할 집합 G의 요소들에 대한 경계 B가 결정되면, 집합 FP는 확률 $P(G)$에 대한 모든 가능한 근사 곱을 구성하기 위해 B의 패턴들을 사용하며 다음은 그림 11의 경계 패턴들($B=\{\{g_2, g_5\}, \{g_3, g_4\}, \{g_1, g_2, g_3\}, \{g_1, g_4, g_5\}\}$)을 이용하여 $G=\{g_1, g_2, g_3, g_4, g_5\}$에 대한 가능한 근사 곱을 구성하는 경우의 예를 나타낸다.

① $\{g_1, g_2, g_3\}, \{g_1, g_4, g_5\} \Rightarrow P(g_1, g_2, g_3) \cdot P(g_4, g_5|g_1)$
② $\{g_1, g_4, g_5\}, \{g_2, g_5\}, \{g_3, g_4\} \Rightarrow$
$P(g_1, g_4, g_5) \cdot P(g_2|g_5) \cdot P(g_3|g_4)$
③ $\{g_2, g_5\}, \{g_3, g_4\}, \{g_1, g_4, g_5\} \Rightarrow$
$P(g_2, g_5) \cdot P(g_3, g_4) \cdot P(g_1|g_4, g_5)$
④ $\{g_1, g_2, g_3\}, \{g_2, g_5\}, \{g_3, g_4\} \Rightarrow$
$P(g_1, g_2, g_3) \cdot P(g_5|g_2) \cdot P(g_4|g_3)$

위의 4가지 근사 곱 구성 경우 외에 $\{g_1, g_2, g_3\}$, $\{g_1, g_4, g_5\}$, $\{g_2, g_5\}$의 조합은 이전 두 패턴이 마지막 패턴인 $\{g_2, g_5\}$을 이미 포함하고 있으므로 적용될 수 없다. 또한 ②, ③ 모두 같은 빈발 패턴들을 사용하고 있으나, 패턴들의 적용 순서가 다르기 때문에 서로 다른 연쇄 규칙을 이용한 근사 곱을 구성한다. 최종적으로 G에 대한 조절 네트워크 구성은 위의 4가지 경우에 대한 연쇄 규칙을 적용하여 가장 최대인 확률(maximal probability)을 가지는 구조를 선택하게 된다. 그러나 이 방법은 단순한(naive) 구성 전략이므로 근사 곱의 구성을 위한 B의 빈발한 유전자 패턴 g의 선택은 [정의 10]의 기준에 의해 선택된다.

〈표 4〉 근사 곱의 구성 예

Covered pattern	Selected pattern	Product approximation using Chain rule	Border list patterns
$\varnothing$	$\varnothing$	$\varnothing$	$\{g_6, g_9, g_{11}\}, \{g_1, g_{11}\}, \{g_2, g_{11}\}, \{g_2, g_6\}$
$\{g_1, g_{11}\}$	$\{g_1, g_{11}\}$	$P(g_1, g_{11})$	$\{g_6, g_9, g_{11}\}, \{g_2, g_{11}\}, \{g_2, g_6\}$
$\{g_1, g_2, g_{11}\}$	$\{g_2, g_{11}\}$	$P(g_1, g_{11})P(g_2 \mid g_{11})$	$\{g_6, g_9, g_{11}\}, \{g_2, g_6\}$
$\{g_1, g_2, g_6, g_{11}\}$	$\{g_2, g_6\}$	$P(g_1, g_{11})P(g_2 \mid g_{11})P(g_6 \mid g_2)$	$\{g_6, g_9, g_{11}\}$
$\{g_1, g_2, g_6, g_9, g_{11}\}$	$\{g_6, g_9, g_{11}\}$	$P(g_1, g_{11})P(g_2 \mid g_{11})P(g_6 \mid g_2)P(g_9 \mid g_6, g_{11})$	$\varnothing$

집합 $G=\{g_1, g_2, \cdots, g_n\}$에 대해, 빈발 패턴들의 경계(*border*), B의 모든 리스트는 [정의 5]의 패턴 응집도(PC)에 대한 내림차순으로 정렬된다. 그런 다음 가장 높은 PC값을 가지는 패턴을 시작으로 해서 점진적으로 근사 곱을 구성해 간다. G에 포함되는 부분 패턴들의 집합을 cov라 할 경우 다음의 선택 규칙을 정의 할 수 있고 알고리즘은 그림 12와 선택 규칙에 대한 프로시저는 그림 13과 같다.

[정의 10] 근사 곱의 구성을 위한 패턴 p의 선택 규칙:

rule 1: $|p-cov| \geq 1$ rule 2: $PC(p) > PC(p')$

rule 3: $length(p) < length(p')$ rule 4: $|p-cov| \leq |p'-cov|$

위의 2~3 규칙에서 패턴 p' 대신 p를 선택한다. ■

여기서 규칙 1은 선택되어질 패턴 p는 반드시 이전에 선택된 패턴에 포함되지 않은 하나 이상의 새로운 항목(유전자)을 포함하는 것을 의미하며 이것은 연쇄 규칙과 근사 곱의 유효성을 보장한다. 규칙 2는 높은 응집도를 가지는 p를 우선한다는 의미이고, 같은 응집도를 갖는 패턴일 경우에는 패턴의 길이가 짧은 것을 선택하게 한다. 즉, 규칙 3은 근사 곱의 구성에서 사용되어지는 패턴의 수를 최대화 시키게 된다. 마지막으로 규칙 4는 남아있는 패턴들 중 이미 포함되지 않은 항목의 수가 최소인 p를 우선한다는 의미이다.

Input: the final set of *FS* containing non-redundant pattern p, a set of genes, G.
Output: border list B and a value of P(T).

1) $B=\{p \in FS \mid p \subset G\}$;
2) *covered* = $\varnothing$; *numerator* = $\varnothing$; *denominator* = $\varnothing$;
3) **for** (i=1 ; *covered* $\subset G$; i++) **do**
4) B_i = **NextCS**(covered, B);
5) *numerator* = *numerator* $\cup B_i$;
6) *denominator* = *denominator* $\cup \{B_i \cap covered\}$;
7) *covered* = *covered* $\cup B_i$;
8) **end for**
9) Output B, *set of denominator* and probability:

$$P(G) = \frac{\prod_{p \subset numerator} P(p)}{\prod_{q \subset denominator} P(q)}$$

(그림 12) 근사 곱의 구성 알고리즘

NextCS(*covered*, B);

$L=\{p \in B \wedge |p-covered| \geq 1\}$;
return $B_i \in L$ such that for all other $B_j \in L$;
1) $PC(B_i) > PC(B_j)$ or
2) $PC(B_i) = PC(B_j)$ and $length(B_i) < length(B_j)$ or
3) $PC(B_i) = PC(B_j)$ and $length(B_i) = length(B_j)$
and $|B_i - covered| \leq |B_j - covered|$;

(그림 13) p 선택 규칙 프로시저

4.2 연쇄 규칙으로부터의 유전자 조절 상호작용 네트워크 구축

4.1절의 근사 곱에 대한 경계 리스트가 결정되면 리스트의 모든 패턴들로부터 조절자(regulator)들을 식별할 수 있다. 네트워크 구성을 위한 유전자 집합 $G=\{g_1, g_2, g_6, g_9, g_{11}\}$이 주어지고, (그림 12)와 (그림 13)의 알고리즘 적용 후의 선택된 경계 B에 대한 리스트와 근사 곱이 주어졌을 경우는 <표 4>이다.

집합 $G=\{g_1, g_2, g_6, g_9, g_{11}\}$에 대해, 연쇄 규칙에 의한 조절자의 식별에 대한 기본 개념은 다음과 같다. 예를 들어 <표 4>의 근사 곱의 구성에서 식(6)을 구할 수 있고,

$$P(G) = P(g_1, g_{11}) \bullet P(g_2 \mid \underline{g_{11}}) \bullet P(g_6 \mid \underline{g_2}) \bullet P(g_9 \mid \underline{g_6, g_{11}}) \quad \text{식(6)}$$

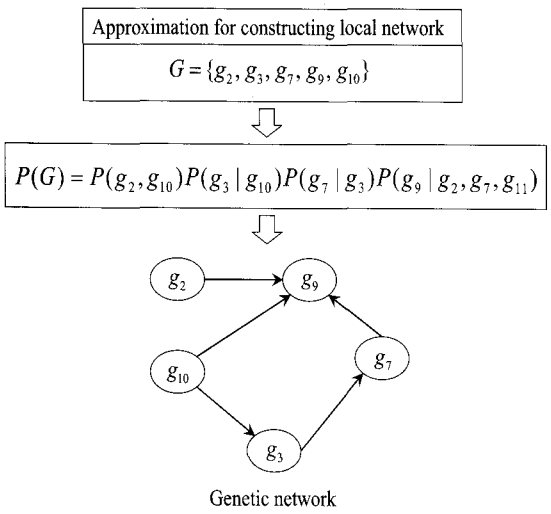
(Regulators)

여기서 조절자는 g_{11}, g_2, g_6이다. 조건부 확률에서의 표현으로 위 식(6)에서 조절자에 대한 각 근사 패턴에 대한 확률값을 다시 표현하면 식(7)이다.

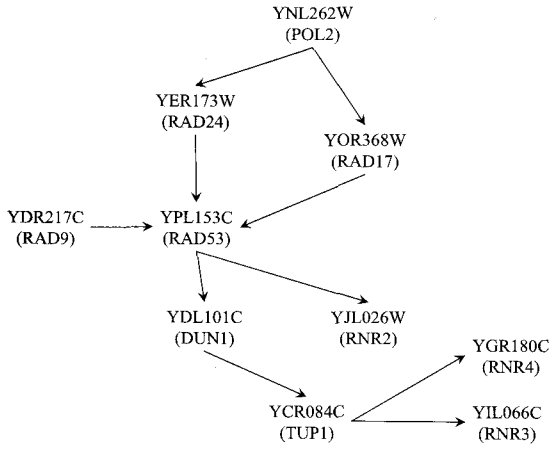
$$\begin{aligned} P(g_2 \mid g_{11}) &= P\left(\frac{g_2 \wedge g_{11}}{g_{11}}\right) = P(g_{11} \rightarrow g_2), \\ P(g_6 \mid g_2) &= P\left(\frac{g_2 \wedge g_6}{g_2}\right) = P(g_6 \rightarrow g_2), \\ P(g_9 \mid g_6, g_{11}) &= P\left(\frac{g_6 \wedge g_9 \wedge g_{11}}{g_6 \wedge g_{11}}\right) = P(g_6, g_{11} \rightarrow g_9) \end{aligned} \quad \text{식(7)}$$

표현식, $P(g_{11} \rightarrow g_2)$는 유전자 g_{11}이 발현한 상태 하에서 g_2가 발현한다는 의미이고, 즉 g_{11}가 g_2의 activator라는 것을 의미한다. $P(g_6, g_{11} \rightarrow g_9)$은 g_6, g_{11}이 유전자 g_9을 조절하는 조절자임을 의미한다. 이러한 방법으로 하여 그림 14와 같이 전체 조절자로 식별 되는 유전자들에 대한 조건부 확률을 구할 수 있고 조절 네트워크를 구축할 수 있다.

예를 들어 DNA damage repair 반응에 참여하는 유전자 집합 *G={RAD9 , RAD17, RAD24, RAD53, POL2, DUN1, CRT1, TUP1, RNR2, RNR3, RNR4}*이 주어질 경우, 상호작용 조절 네트워크는 (그림 15)이다. (그림 15)에서의 유전자 RAD53은 DNA repair 프로세스의 중요한 조절자 중의 하나이고[16] RNR2, RNR3, RNR4 유전자들은 DNA 합성 및 복원에 관련된 유전자들이며 DNA repair 프로세스 시작을 위한 트리거이다.



(그림 14) 유전자 조절 네트워크 구성 예



(그림 15) DNA damage repair 반응에 관련된 유전자들의 조절 네트워크

5. 실험 및 평가

유전자 조절 상호작용 네트워크 구축에 대한 실험은 saccharomyces cerevisiae[9]과 Yeast Protein Database(YPD)[17]에서 검색된 유전자들로서 조절 관계(activation, inhibition)를 실험에 사용하였다. saccharomyces cerevisiae의 성장 단계 중 alpha-factor, cdc28 동기화 단계에 해당되는 유전자들과 일치하는 activation, inhibition 관계는 <표 5>와 같다.

<표 5> YPD 데이터베이스와 saccharomyces cerevisiae 데이터로부터의 생성된 두 성장 단계에서의 조절 관계 수

Data set	# of genes	# of activation	# of inhibition
alpha-factor	332	343	96
cdc28	365	469	155

alpha-factor 데이터 집합은 332개의 유전자만이 총 18 time-point에 대해서 매치 되었고 이 유전자들은 343개의 기능이 알려진 activation 관계, 96개의 inhibition 관계에 포함된다. cdc28 데이터에는 총 365개의 유전자가 매치되며 121 유전자들이 일치되지 않아 제외하였다. cdc28에는 각각 매치되는 469, 155개의 조절 관계를 포함한다.

발현 비율에 대한 이산화는 positive 변형과 negative 변형에 대해 두 가지의 데이터 집합을 생성한다. positive 데이터 변환의 경우 gene-up을 1(gene-down=0, unchange=0)로 하고 negative의 경우는 gene-down을 1(gene-up=0, unchange=0)로 하였다.

연쇄 규칙에 의해 조절자로 발견된 유전자들은 이미 알려진 생물학적 기능을 가지는 유전자들로 식별하였다. <표 6>은 제안된 알고리즘 적용 후에 cell cycle 데이터로부터의 조절자로 예측된 기능이 알려진 유전자들과 조절자로 예측은 되었으나 아직 annotate 되지 않은 유전자와 식별되지 않은 ORF들의 예를 보여준다. <표 6>에서 조절자로 예측된 대부분의 유전자들은 촉매 기능을 가지며, 많은 유전자들은 단백질 합성 프로세스에 포함된다. 단백질 합성은 세포 안에서 활성 프로세스들 중의 하나이며, 실제로 모든 형태의 cellular function을 필요로 한다. 따라서 단백질 생합성에 포함된 유전자들은 조절자로 예측되어질 수 있다.

실험은 alpha-factor와 cdc28 데이터 집합에서 발현되는 모든 빈발 유전자 발현 패턴들을 찾아내고 연쇄 규칙을 적용하여 조절자들을 예측한다. 예측에 대한 검증은 <표 7>의 confusion matrix로 표현하며, 예측 결과의 평가는 Recall과 Precision 그리고 F-Measure와 평균절대오차(MAE: Mean Absolute Error)를 이용하였다. <표 5>의 alpha-fator와 cdc28 데이터 집합의 activator/inhibitor 예측에 대한 결과는 <표 8>과 같다.

$$recall, r=\frac{TP}{TP+FN}, \quad \text{식(8)}$$

$$precision, p=\frac{TP}{TP+FP} \quad \text{식(9)}$$

$$F-Measure, f=\frac{2 \cdot precision \cdot recall}{precision+recall} \quad \text{식(10)}$$

$$MAE=|E|\frac{\sum_{i=0}^{N}|\varepsilon_i|}{N} \quad \text{식(11)}$$

〈표 6〉 조절자로 예측된 유전자의 예

ORF	Gene	Biological Function	Molecular Function
YIL123W	SIM1	cell cycle	molecular_function unknown
YLR075W	RPL10	protein biosynthesis	structural protein of ribosome
YGL031C	RPL24A	protein biosynthesis	structural protein of ribosome
YLR325C	RPL38	protein biosynthesis	structural protein of ribosome
YGR148C	RPL24B	protein biosynthesis	structural protein of ribosome
YER102W	RPS8B	protein biosynthesis	structural protein of ribosome
YJR099W	YUH1	Deubiquitylation	ubiquitin-specific protease
YLR167W	RPS31	protein biosynthesis	structural protein of ribosome
YFR052W	RPN12	ubiquitin-dependent protein degradation	succinate--CoA ligase(ADP-forming)
YHL015W	RPS20	protein biosynthesis	structural protein of ribosome
YKL145W	RPT1	ubiquitin-dependent protein degradation	adenosinetriphosphatase
YOR157C	PUP1	ubiquitin-dependent protein degradation	multicatalytic endopeptidase
YCR034W	FEN1	fatty acid biosynthesis	molecular_function unknown
YER074W	RPS24A	protein biosynthesis	structural protein of ribosome
YHR006W	STP2	tRNA splicing	molecular_function unknown
YKL003C	MRP17	protein biosynthesis	structural protein of ribosome
YGL103W	RPL28	protein biosynthesis	structural protein of ribosome
YOL139C	CDC33	protein synthesis initiation	translation initiation factor
YML063W	RPS1B	protein biosynthesis	structural protein of ribosome
YNL315C	ATP11	protein complex assembly	chaperone
YJL092W	HPR5	DNA repair	A helicase
YIL062C	ARC15	cell growth or maintenance	structural protein
...	...	...	...

ORF	Gene
YDR041W	RSM10
YIL131C	FKH1
YGL097W	SRM1
YPR100W	
YLR083C	EMP70
YJL183W	MNN11
YMR215W	
YMR311C	GLC8
YBR086C	IST2
YKL195W	
YLR217W	
YOR246C	
YGL139W	
YKL169C	
YLL032C	
YPR100W	
YLL044W	
YHR097C	
YNL087W	
YJL099W	

〈표 7〉 조절자 예측 평가를 위한 confusion matrix

		Predicted regulator	
		regulator	non-regulator
Actual regulator	regulator	TP	FN
	non-regulator	FP	TN

〈표 8〉 조절자 예측에 대한 결과 요약

	TP rate	FP rate	Precision	Recall	F-Measure	Prediction	MAE
alpha factor	0.735	0.255	0.750	0.745	0.742	regulator	0.3451
	0.745	0.265	0.729	0.735	0.737	non-regulator	
cdc28	0.576	0.138	0.814	0.576	0.675	regulator	0.3144
	0.862	0.424	0.659	0.862	0.747	non-regulator	

6. 결 론

이 논문에서는 유전자 조절 네트워크 구축을 통해 다른 유전자들의 발현 레벨을 조절하는 조절자(activator, inhibitor)를 예측하였다. 이를 위해서 먼저, 유전자 발현 데이터를 발현 비율에 기반한 3가지 항목으로 표현하여 빈발 패턴 마이닝이 적용 가능하도록 트랜잭션화 하였다. 제안된 유전자 조절 네트워크의 구축 과정은 첫째, 전처리된 각 유전자 발현 데이터에서 빈발한 유전자 패턴들의 발견한다. 이 과정에서 효율적인 패턴 탐사를 위해 FP-growth 알고리즘을 적용하였고, 패턴의 새로운 유용성 측정 지표의 정의 그리고 압축 패턴 트리를 이용한 대량의 중복 패턴 제거 방법을 제시하였다. 마지막 단계에서는 중복 패턴이 제거된 유전자 패턴들로부터 연쇄 규칙을 이용하여 네트워크를 확률적 모델로 추정하였다. 실험은 *saccharomyces cerevisiae* 의 성장 주기 중 alpha-factor와 cdc28 데이터 집합에 대해서 유전자의 조절 레벨을 단순한 이진 변환(up, down)보다 더 잘 반영할 수 있는 positive, negative 데이터 변환을 하여 실험하였다. 또한 실험 결과는 각각의 데이터 집합에 대해, 예측된 조절자와 이미 알려진 결과와의 비교를 통해 검증하였다.

참 고 문 헌

[1] Friedman, N., Linial, M., Nachman, I. and Pe'er, D., "Using Bayesian networks to analyze expression data", Journal of Computational Biology, 7:601-620, 2000.

[2] Husmeier, D., "Sensitivity and specificity of inferring genetic regulatory interactions from microarray experiments with dynamic Bayesian networks", Bioinformatics, 19:2271-2282, 2003.

[3] Ting Chen, Vladimir Filkov, Steven S. Skiena, "Identifying Gene Regulatory Networks from Experimental Data", RECOMB, 94-103, 1999.

[4] Van Someren, E. P., Wessels, L. F. A., and Reinders, "Linear modeling of genetic networks from experimental data. Proc., ISMB, 355-366, 2000.

[5] Holter, N. S., Maritan, A., Fedoroff, N. V. and Banavar, J. R., "Dynamic modeling of gene expression data, Proc., Natl, Acad. Sci. 1693-1698, 2000.

[6] Rishi Khan, Yujing Zeng, Javier Garcia-Frias and Guang Gao, "A Bayesian Modeling Framework for Genetic Regulation", Proc., CSB'02, 2002.

[7] Akutsu, T., Miyano, S., and kuhara, S., "Identification of genetic networks from a small number of gene expression patterns under the Boolean network model", Pacific Symposium on Biocomputing 17-28, 1999.

[8] Eisen, M. B., Spellman, P. T., Brown, P. O., and Botstein, D., "Cluster Analysis and Display of Genome-Wide Expression Patterns". Proc., National Academy of Science. 95:14863-14868, 1998.

[9] Spellman, P. T., Sherlock, G., Zhang, M. Q., Iyer, V. R., Anders, K., Eisen, M. B., Brown, P. O., Botstein, D. and Futcher, B., "Comprehensive identification of cell cycle-regulated genes of the yeast Saccharomyces cerevisiae by microarray hybridization", Molecular Biology of the Cell, 9:3273-3297. 1998.

[10] Tamayo, P., Slonim, D., Mesirov, J., Zhu, Q., Kitareewan, S., Dmitrovsky, E., Lander, E. and Golub, T. "Interpreting patterns of gene expression with selforganizing maps". PNAS, 96:2907-2912. 1999.

[11] Brown, M. P., Grundy, W. N., Lin, D., Sugnet, C. W., Furey, T. S., Ares Jr., and Haussler, D., "Knowledge-based analysis of microarray gene expression data by using support vector machines". PNAS, 4:97(1):262-7. 2000.

[12] Han, J., Pei, J., Yin, Y., "Mining frequent patterns without candidate generation". In SIGMOD'00, Dallas, TX, 2000.

[13] Forsyth, R. and Rada, R., "Machine Learning applications in Expert Systems and Information Retrieval", Ellis Horwood Limited, 1986.

[14] Li, W., Han, J. and Pei, J., "CMAR: Accurate and Efficient Classification Based on Multiple Association Rules", Proc., Interna'l Conf. on Data Mining, 2001.

[15] Meretakis, D. and Wuthrich, B., "Extending naïve bayes classifiers using long itemsets", Proc., the 5th ACM SIGKDD Conference on Knowledge Discovery and Data Min-ing, 165-174, 1999.

[16] Elledge, S. J. and Davis, R. W., "Identification of the DNA damage-responsive element of RNR2 and evidence that four distinct cellular factors bind it", Molecular and Cell Biology, 9(12):5373-86. 1989.

[17] Yeast Protein Database (YPD) (http://www.proteome.com)

이 헌 규

e-mail : hglee@dblab.chungbuk.ac.kr

2002년 경기대학교 전자계산학과(학사)

2004년 충북대학교 전자계산학과 (이학석사)

2004년~2006년 한국표준과학연구원 위촉연구원

2004년~현재 충북대학교 전자계산학과 박사과정

관심분야 : 시공간 데이터베이스, 데이터마이닝, 유비쿼터스 컴퓨팅, 바이오인포매틱스 등

류 근 호

e-mail : khryu@dblab.chungbuk.ac.kr
1976년 숭실대학교 전산학과(학사)
1980년 연세대학교 공업대학원 전산전공(공학석사)
1988년 연세대학교 대학원 전산전공 (공학박사)
1976년~1986년 육군 군수 지원사 전산실(ROTC 장교), 한국전자통신연구원(연구원), 한국방송통신대학교 전산학과(조교수) 근무
1989년~1991년 University of Arizona, Research Staff (TempIS 연구원, Temporal DB)
1986년~현재 충북대학교 전기전자 컴퓨터공학부 교수.
관심분야 : 시간 데이터베이스, 시공간 데이터베이스, Temporal GIS, 지식기반 정보검색 시스템, 유비쿼터스컴퓨팅 및 스트림데이터처리, 데이터 마이닝, 데이터베이스 보안, 바이오인포매틱스

정 두 영

e-mail : fiorgeo@trut.cbnu.ac.kr
2001년 서강대학교 컴퓨터과학과(공학박사)
1987년~현재 충북대학교 전기전자 컴퓨터공학부 교수
관심분야 : 데이터 통신, 데이터베이스, 데이터 망