# 단백질 상호작용 네트워크 및 유전자 발현값을 이용한 중복 허용 단백질 복합체 탐색 방법

## (Detecting Overlapping Protein Complexes Employing Protein Interaction Network and Gene Expression Data)

안 재 균 † 여 윤 구 † 윤 영 미 †† 박 상 현 †††
(Jaegyoon Ahn) (Yunku Yeu) (Youngmi Yoon) (Sanghyun Park)

**요 약** 단백질 복합체(protein complex)를 찾아내는 것은 생물학적인 현상들을 이해하는 데 있어서 가장 기본적으로 선행되어야 할 과제 중 하나이다. 단백질 복합체를 찾는 방법 중 가장 널리 쓰이는 방법은 단백질 상호 작용 네트워크(protein interaction network)의 군집화(clustering)를 이용하는 것이다. 그러나 이러한 방법을 이용할 경우 단백질 상호 작용 네트워크의 각 간선(protein interaction)은 높은 거짓 긍정(false positive) 및 거짓 부정(false negative) 오류율을 보이기 때문에, 네트워크로부터 단백질 복합체를 정확히 찾아내는 것은 어려운 작업이다. 따라서 본 연구에서는 네트워크 데이터 외에도 유전자 발현값 데이터를 추가적으로 이용해서 단백질 복합체를 찾는 방법을 제시한다. 이 방법은 어떤 단백질은 여러 단백질 복합체에 속할 수 있으므로, 중복을 허용하는 네트워크 탐색 방법을 사용한다. 결과적으로, 본 연구에서 제시한 방법을 이용했을 경우 기존의 단백질 복합체 탐색 방법보다 정확하게 단백질 복합체를 찾음을 확인할 수 있었다.

**키워드** : 데이터 마이닝, 네트워크 클러스터링, 단백질 복합체, 단백질 상호 작용 네트워크

***Abstract*** Detecting protein complexes is essential work for understanding biological functions and processes. Most protein complexes detecting methods are based on clustering the protein interaction network. However, one of the difficulties in these methods originates from the fact that protein interactions suffer from high false positive rate. We propose a protein complex detecting algorithm which employs gene expression data, as well as protein interaction network. The proposed algorithm allows overlapping of the protein complexes based on the fact that some proteins can be involved in several complexes at the same time. As a result, we could confirm that our algorithm is more accurate than existing algorithms.

**Key words** : data mining, network clustering, protein complex, protein interaction network

· 이 논문은 2012년도 정부(교육과학기술부)의 재원으로 한국연구재단의 기초연구사업 지원을 받아 수행된 것임(2012-0002887)

† 학생회원 : 연세대학교 컴퓨터과학과
ajk@cs.yonsei.ac.kr
yyk@cs.yonsei.ac.kr
†† 종신회원 : 가천대학교 컴퓨터공학과 교수
ymyoon@gachon.ac.kr
††† 종신회원 : 연세대학교 컴퓨터과학과 교수
sanghyun@cs.yonsei.ac.kr
(Corresponding author임)

논문접수 : 2011년 10월 26일
심사완료 : 2011년 12월 22일


정보과학회논문지 : 데이타베이스 제39권 제3호(2012.6)

## 1. 서 론

대부분의 단백질은 다른 단백질들과 함께 복합체(complex)를 이루면서 복잡한 세포내 기능에 관여한다고 알려져 있다[1]. 따라서 단백질 복합체를 찾는 것은 생물학적 기능이나 세포내 현상들을 이해함에 있어서 기본적이고 중요한 일이다.

단백질 복합체를 찾기 위해서 널리 쓰이는 방법들은 대부분 단백질 상호작용 네트워크(protein interaction network, 이하 PIN)의 탐색에 기본을 둔다. PIN은 정점(node)을 단백질로 하고, 물리적으로 결합하는 두 단백질 정점 간에 간선(edge)이 존재하는 무방향성 그래프로 모델링할 수 있다. PIN을 구축함에 있어서 가장

중요한 것은 물리적으로 결합하는 두 단백질을 찾아내는 것이며, 이것을 단백질 상호작용(protein-protein interaction, 이하 PPI)이라고 한다. 이러한 상호작용을 찾을 수 있는 대표적인 실험적 방법으로 yeast two-hybrid system[2]과 Mass Spectrometry[3]를 들 수 있다. 최근 이러한 방법의 발전으로 풍부한 양의 정확한 PPI 데이터를 얻을 수 있었으며, 결과적으로 양질의 PIN의 구축이 가능해졌다.

단백질 복합체가 서로 상호작용하는 단백질들의 집합이기 때문에, 단백질 복합체는 대부분의 경우 네트워크 내에서 서로 가까운 거리에 있는 단백질의 집합이거나, 완전 그래프에 가까운 부분 네트워크의 형태로 나타난다. 즉, PIN에서 단백질 복합체를 찾기 위해서는 네트워크에서 이러한 특징을 가진 부분 네트워크를 찾아내는 알고리즘을 이용할 수 있으며, 이러한 알고리즘들을 통상적으로 네트워크 클러스터링 알고리즘이라 한다.

네트워크 클러스터링 알고리즘은 이미 많이 연구 개발된 바 있다. MCODE[4]는 연결이 많이 되어 있는 정점에 가중치를 부여하고, 가중치가 높은 정점을 출발점(seed)로 해서 지역적 네트워크 탐색을 통해서 높은 가중치를 가진 정점을 포함하는 방식으로 부분 네트워크를 찾는다. MCODE의 단점은, 결과 부분 네트워크가 연결 강도가 높은 정점들로 이루어져 있을 뿐, 실제로 서로 간에 강하게 연결되어 있다는 보장이 없다는 것이다. Markov clustering algorithm(MCL)[5]은 그래프 내에서 강하게 연결된 부분과 약하게 연결된 부분을 구분함으로써 그래프를 조각내는 방식을 취한다. MCL은 최근까지도 매우 우수한 성능을 보임이 증명된 바 있으며[6,7], 많은 방법으로 변형, 발전되었다.

앞서 기술된 네트워크 클러스터링 알고리즘들은 공통적으로 결과로 도출되는 부분 네트워크들의 중복을 허용하지 않는다는 특징을 보인다. 즉, 한 정점은 단 하나의 부분 네트워크에만 속할 수 있다. 그러나 실제로 어떤 단백질은 여러 단백질 복합체에 속할 수 있다[8]. 따라서 단백질의 중복을 허용하는 단백질 복합체가 보다 현실적이라고 할 수 있으며, 중복을 허용하는 네트워크 클러스터링 알고리즘을 이용할 경우 보다 정확한 단백질 복합체를 찾을 수 있다. 최근 이러한 네트워크 클러스터링 알고리즘도 많이 연구되고 있다.

DPClus[9]는 초기 정점(seed)으로부터 시작해서, 부분 그래프에서 간선의 밀도가 일정 정도 이상이 되도록 유지하면서 주변의 정점들을 흡수하는 방식으로써 중복을 허용하지 않는 단백질 복합체를 우선적으로 찾아낸다. 그 후 이러한 복합체와 연결된 단백질을 포함시켜 나가는 방식을 이용해서 중복을 허용하는 단백질 복합체를 찾는다. CFinder[10]는 Clique Percolation Method (CPM)[11]에 기반을 두어 중복을 허용하는 단백질 복합체를 찾는 알고리즘이다. CPM은 단백질 복합체를 서로 간에 (k-1)개의 정점을 공유하는 k-clique의 합집합으로 정의한다. CFinder의 결과는 k 값에 민감한데, k가 클수록 작은 크기의 높은 간선 밀도를 가지는 부분 네트워크를 찾는 경향이 있다. [12]는 먼저 네트워크의 간선을 가상의 정점으로 치환하고, 정점을 공유하는 간선(가상의 정점) 간에 새로운 간선을 만드는 방법으로 네트워크를 변환시킨다. 변환된 네트워크의 가상의 정점들은 서로 간에 연결 강도가 클수록 거리가 가까우며, 이를 이용해서 계층적 클러스터링(hierarchical clustering) 기법을 적용함으로써, 가상의 정점들을 클러스터링할 수 있다. 이때, 가상의 정점은 원래 네트워크의 간선에 해당하므로, 결과 부분 네트워크는 결과적으로 간선의 집합을 의미하며, 결과적으로 부분 네트워크들은 정점을 공유할 수 있다.

중복을 허용하거나 허용하지 않는 기존 네트워크 클러스터링 알고리즘들의 결과는 공통적으로 PIN에 상당히 종속적이다. 그러나 PIN을 이루는 PPI는 높은 거짓 긍정(false positive) 및 거짓 부정(false negative) 오류율을 보이는 것으로 알려져 있으며[13,14], 따라서 결과적으로 탐색된 단백질 복합체 또한 높은 오류율을 보일 수밖에 없다. 본 논문에서는 유전자 mRNA 발현 데이터를 이용해서 단백질 복합체 검색의 정확도를 높이고자 한다. 유전자의 mRNA 발현값은 단백질의 양과 정확한 상관관계를 이루지는 않는다. 하지만, 같은 기능에 관련되어 있는 단백질 집합은 그 mRNA 발현값이 동시에 높거나, 동시에 낮은 경향을 보인다[15]. 본 논문에서는 이러한 특징을 이용함으로서 PIN의 불완전성을 보완했으며, 동시에 중복을 허용하는 새로운 네트워크 탐색 방법을 도입해서 새로운 단백질 복합체 탐색 방법을 제시한다. 이 방법을 생물학적인 검증이 풍부하게 이루어진 Saccharomyces cerevisiae(yeast)의 PIN 및 유전자 발현 데이터에 적용시킨 결과, 기존의 방법들보다 정확하게 단백질 복합체를 찾을 수 있음을 확인할 수 있었다.

본 논문의 구성은 다음과 같다. 2장에서는 본 논문에서 제안하는 단백질 복합체 탐색 알고리즘을 설명한다. 3장에서는 실험을 통하여 제안한 알고리즘의 성능을 평가한다. 마지막으로 4장에서는 본 논문을 요약하고 앞으로의 연구 방향을 제시한다.

## 2. 단백질 복합체 탐색 기법

본 논문에서 제시하는 단백질 복합체 탐색 알고리즘은 크게 두 단계로 이루어진다. 먼저, PIN의 각 간선을 유전자 발현 데이터를 이용해서 스코어링하는 단계가 선행된다. 이후 스코어링된 각 간선을 이용해서 네트워

크를 탐색하면서 단백질 복합체를 찾아낸다.

### 2.1 기호의 정의

네트워크 간선의 스코어링 방법 및 탐색 알고리즘을 설명하기에 앞서서 표 1에 몇 가지 기호를 정의한다.

### 2.2 간선의 스코어링 기법

PIN의 간선은 물리적으로 상호작용하는 두 단백질인 PPI로 구성된다. 그러나 실제 두 단백질이 상호작용을 하는지의 여부는 실험적으로 증명되지 않은 경우가 많다. 즉 PIN은 높은 오류율을 보인다[13,14]. 이러한 오류율로 인한 정확도의 감소를 막기 위해서, 본 논문에서는 유전자 mRNA 마이크로어레이 실험 데이터를 추가적으로 사용한다. 이 데이터는 여러 조건 하에서 유전자 집합의 mRNA 발현값을 나타내는 2차원 데이터이다. 그림 1-1)은 8개의 조건($c_0$~$c_7$) 하에서 5개의 유전자($g_0$~$g_4$)의 mRNA 발현값을 보여준다. 이 예제에서 한 유전자는 8개의 발현값을 가지며, 본 논문에서는 이러한 발현값의 집합을 발현값 벡터라고 부른다.

각 간선을 이루는 두 개의 노드는 같은 기능에 관여되어 있는 단백질일 가능성이 높다. 그런데, 이러한 경우, 두 단백질을 번역(translation)하는 두 유전자의 mRNA 발현값 또한 높은 상관관계를 가지고 있다[15]. 이 상관관계를 알아보기 위해서 흔히 사용되는 방법은 Pearson Correlation Coefficient(PCC)와 같은 상관 계수이다. 두 유전자의 발현값 벡터들의 PCC가 1에 가까울수록 두 유전자는 강한 양의 상관관계를 가지며, PCC가 -1에 가까울수록 두 유전자는 강한 음의 상관관계를 가진다고 한다.

PCC 등의 상관계수를 구하는 과정에서는 모든 조건 하에서의 유전자 발현값을 고려한다. 하지만, 실제로 두 유전자 혹은 단백질은 세포내에서 특정 조건 하에서만 같은 기능에 관여하는 경우가 많다. 따라서 모든 조건 하에서의 발현값의 상관계수는 실제의 상관관계를 반영하지 못할 수 있다. 본 연구에서는 이와 같은 단점을 해결하기 위해서, 새로운 상관관계 측정 방법을 개발했다. 이를 위해서 먼저, 그림 1-1)과 같은 유전자 발현값 데이터에서 각 유전자 발현값 벡터를 k-means clustering을 이용해서 클러스터링한다. 이 때, k를 3으로 함으로써, 벡터를 값에 따라서 3개의 그룹으로 나눌 수 있으며, 가장 값이 큰 그룹에 속한 발현값을 1로, 중간 그룹에 속한 발현값을 0으로, 가장 낮은 그룹에 속한 발현값을 -1로 변환가능하다. 그림 1-2)는 그 결과를 보여준다.

표 1 네트워크 탐색 알고리즘 기호

| 기호 | 설명 |
|---|---|
| $exp(g_i, c_j)$ | 조건 $c_j$ 하에서 유전자 $g_i$ 의 발현값(expression) |
| $PC$ | 임의의 단백질 복합체(protein complex)를 이루는 단백질의 집합 |
| $R$ | 이미 찾아진 $PC$의 집합, 초기값은 ∅ |
| $PC_{cur}$ | 현재 탐색하고 있는 단백질 복합체를 이루는 단백질의 집합 |
| $Next_{cur}$ | $PC_{cur}$ 의 각 단백질과 간선으로 직접 연결된 단백질의 집합 |
| $CC\_thre$ | clustering coefficient($CC$)값의 최소 허용 유지율 (사용자 입력값) |
| $SCS\_thre$ | $\|SCS\|$값의 최소 허용 유지율 (사용자 입력값) |

1)

| condition / gene | $c_0$ | $c_1$ | $c_2$ | $c_3$ | $c_4$ | $c_5$ | $c_6$ | $c_7$ |
|---|---|---|---|---|---|---|---|---|
| YAL003W ($g_0$) | 0.9168 | -0.1817 | -1.2823 | -0.9802 | -2.4850 | 1.4109 | 1.5783 | -0.3303 |
| YAL012W ($g_1$) | 1.2462 | -1.0148 | -2.2439 | -1.9135 | -1.4585 | 1.5536 | -1.2155 | -0.4439 |
| YAL014C ($g_2$) | 1.0714 | -0.2817 | -1.4024 | 0.1978 | -0.9318 | 1.7301 | -0.2687 | 1.5962 |
| YAL015C ($g_3$) | -0.2791 | -0.9459 | 1.1808 | 0.4089 | 1.1626 | 0.9933 | 0.1190 | -2.0700 |
| YAL016W ($g_4$) | 2.1972 | 0.0445 | 1.2325 | -1.5092 | 1.8840 | -2.3301 | 0.6301 | -0.5301 |

K-means clustering (k=3)

2)

| condition / gene | $c_0$ | $c_1$ | $c_2$ | $c_3$ | $c_4$ | $c_5$ | $c_6$ | $c_7$ |
|---|---|---|---|---|---|---|---|---|
| YAL003W ($g_0$) | 1 | 0 | -1 | -1 | -1 | 1 | 1 | 0 |
| YAL012W ($g_1$) | 1 | -1 | -1 | -1 | -1 | 1 | -1 | 0 |
| YAL014C ($g_2$) | 1 | 0 | -1 | 0 | -1 | 1 | 0 | 1 |
| YAL015C ($g_3$) | 0 | -1 | 1 | 0 | 1 | 1 | 0 | -1 |
| YAL016W ($g_4$) | 1 | 0 | 1 | -1 | 1 | -1 | 0 | 0 |

그림 1 1) 5개의 유전자의 8개의 조건 하에서 mRNA 발현값, 2) 각 유전자 발현값 벡터를 k-means clustering (k=3) 방법으로 클러스터링한 결과

두 유전자 $g_i$ 및 $g_j$의 mRNA 발현값이 동시에 높거나 동시에 낮은 조건의 수가 많을수록 두 유전자는 깊은 양의 상관관계에 있다고 할 수 있을 것이다. 이러한 조건의 집합을 $P$라고 하며, 다음 식 (1)과 같이 정의된다.

$$P(g_i, g_j) = \{c \mid (\exp(g_i, c) = 1 \text{ and } \exp(g_j, c) = 1) \text{ or } (\exp(g_i, c) = -1 \text{ and } \exp(g_j, c) = -1)\} \quad (1)$$

또한 두 유전자 $g_i$ 및 $g_j$의 mRNA 발현값이 하나는 높은데 하나는 낮은 조건의 수가 많을수록 두 유전자는 깊은 음의 상관관계에 있다고 할 수 있다. 이러한 조건의 집합을 $N$이라고 하며, 다음 식 (2)와 같이 정의된다.

$$N(g_i, g_j) = \{c \mid (\exp(g_i, c) = 1 \text{ and } \exp(g_j, c) = -1) \text{ or } (\exp(g_i, c) = 1 \text{ and } \exp(g_j, c) = -1)\} \quad (2)$$

두 유전자 $g_i$ 및 $g_j$에 대해서 조건의 집합 $SCST$ (Similar Condition Set for Two genes)은 다음의 식 (3)과 같이 계산된다.

$$SCST(\{g_i, g_j\}) = \begin{cases} \text{if } |P(g_i, g_j)| \geq |N(g_i, g_j)|, & P(g_i, g_j) \\ else & N(g_i, g_j) \end{cases} \quad (3)$$

예를 들어서, 그림 1-2)의 유전자 $g_0$ 및 $g_1$의 $SCST$ ($\{g_0, g_1\}$)의 경우 $P$ = {$c_0$, $c_2$, $c_3$, $c_4$, $c_5$}이고 $N$ = {$c_6$}이므로, 식 (3)에 의해서 결과는 {$c_0$, $c_2$, $c_3$, $c_4$, $c_5$}이다. 식 (3)을 통해서 PIN의 모든 간선에 대해서, 간선을 구성하는 두 단백질(유전자) 사이의 $SCST$를 구할 수 있다. 이때, 임의의 간선 $e$ = ($g_i$, $g_j$)의 스코어는 |$SCST$ ($\{g_0, g_1\}$)|로 정의할 수 있다.

### 2.3 네트워크의 탐색

2.2절에서와 같이 PIN의 모든 간선에 대한 스코어링이 완료된 후, 모든 간선들을 스코어로 내림차순 정렬해서 순서대로 네트워크 탐색의 시작점으로 한다. 즉, 네트워크의 탐색은 최대 네트워크의 간선의 수만큼 이루어진다. 단, 시작점이 되는 간선 ($g_i$, $g_j$)가 주어졌을 때, $g_i$ 및 $g_j$가 $R$ 중 하나 이상의 $PC$에 속한다면, 이 시작점에 대해서는 네트워크 탐색을 진행하지 않는다.

시작점이 되는 간선 ($g_i$, $g_j$)에 대한 네트워크 탐색의 시작 전에, $PC_{cur}$를 $g_i$ 및 $g_j$로 초기화하고, $Next_{cur}$를 $g_i$ 및 $g_j$와 직접 연결된 단백질들로 초기화한다. 이렇게 초기화된 $PC_{cur}$ 및 $Next_{cur}$에 대해서 네트워크의 탐색은 다음의 두 가지 단계로 구성된다.

**단계 1.** $Next_{cur}$의 각 단백질 $p$를 $PC_{cur}$에 추가해서 $PC_{cur}'$를 만든다. 이때, 식 (4)에 의해서 $CC(PC_{cur}')$를 구한다.

$$CC(PC_{cur}') = \frac{2 \times e}{n \times (n-1)} \quad (4)$$

, where $e = PC_{cur}'$의 간선의 수, $n = PC_{cur}'$의 정점의 수

식 (4)는 부분 네트워크의 간선의 수를 부분 네트워크가 가질 수 있는 간선의 최대 수로 나눈 값으로써, 부분 네트워크의 간선의 밀도가 높을수록 커지는 값임을 알 수 있다.

또한, $PC_{cur}$의 임의의 단백질 $g_i$와 추가하려는 단백질 $p$의 발현값이 동시에 1이거나 -1인 조건 집합 $P$는 다음 식 (5)와 같이 정의된다.

$$P(g_i, p) = \{c \mid (\exp(g_i, c) = 1 \text{ and } \exp(p, c) = 1) \text{ or } (\exp(g_i, c) = -1 \text{ and } \exp(p, c) = -1)\} \quad (5)$$

$PC_{cur}$의 임의의 단백질 $g_i$와 추가하려는 단백질 $p$의 발현값이 하나는 1이고 다른 하나는 -1인 조건 집합 $N$은 식 (6)과 같이 정의된다.

$$N(g_i, p) = \{c \mid (\exp(g_i, c) = 1 \text{ and } \exp(p, c) = -1) \text{ or } (\exp(g_i, c) = 1 \text{ and } \exp(p, c) = -1)\} \quad (6)$$

이 때, $PC_{cur}'$는 세 개 이상의 단백질로 이루어진 집합이며, $PC_{cur}'$의 $SCSM$ (Similar Condition Set for Multiple genes)은 다음의 식 (7)에 의해서 구한다.

$$SCSM(PC_{cur}') = \begin{cases} \text{if } |P'| \geq |N'|, & P' \\ else & N' \end{cases} \quad (7)$$

$$, where\ P' = \begin{cases} SCSM(PC_{cur}) \cap P, & \text{if } |PC_{cur}| > 2 \\ SCST(PC_{cur}) \cap P, & \text{if } |PC_{cur}| = 2 \end{cases}$$

$$, \text{and } N' = \begin{cases} SCSM(PC_{cur}) \cap N, & \text{if } |PC_{cur}| > 2 \\ SCST(PC_{cur}) \cap N, & \text{if } |PC_{cur}| = 2 \end{cases}$$

예를 들어서, 그림 1-2)의 유전자 $g_0$ 및 $g_1$의 $SCST$ ($\{g_0, g_1\}$)은 식 (3)에 의해서 {$c_0$, $c_2$, $c_3$, $c_4$, $c_5$}이다. {$g_0$, $g_1$}에 $g_2$를 추가해서 $PC_{cur}'$ = {$g_0$, $g_1$, $g_2$}가 된 경우, $P'$ = {$c_0$, $c_2$, $c_4$, $c_5$}이고 $N' = \varnothing$이므로, $SCSM(PC_{cur}')$ = {$c_0$, $c_2$, $c_4$, $c_5$}이다. 마찬가지로, {$g_0$, $g_1$, $g_2$}에 $g_3$을 추가해서 $PC_{cur}'$ = {$g_0$, $g_1$, $g_2$, $g_3$}이 된 경우 $SCSM(PC_{cur}')$ 또한 식 (7)으로 계산 가능하며, 결과는 {$c_2$, $c_4$}이다.

$CC(PC_{cur}')$와 $SCSM(PC_{cur}')$를 구했다면, |$SCS$| 및 $CC$의 유지율을 각각 식 (8) 및 식 (9)로 나타낼 수 있으며, $PC_{cur}$에 추가된 $p$의 스코어 $score(p)$는 식 (10)과 같이 정의된다.

$$|SCS| \text{의 유지율} = \begin{cases} |SCSM(PC_{cur}')|/|SCSM(PC_{cur})|, & \text{if } |PC_{cur}| > 2 \\ |SCSM(PC_{cur}')|/|SCST(PC_{cur})|, & \text{if } |PC_{cur}| = 2 \end{cases} \quad (8)$$

$$CC\text{의 유지율} = CC(PC_{cur}')/CC(PC_{cur}) \quad (9)$$

$$score(p) = |SCS| \text{의 유지율} \times CC\text{의 유지율} \quad (10)$$

**단계 2.** $Next_{cur}$의 단백질 중 |$SCS$|의 유지율이 $SCS_thre$보다 작은 단백질이나 $CC$의 유지율이 $CC_thre$보다 작은 단백질은 $Next_{cur}$에서 삭제된다. 그리고 삭제되지 않은 단백질 중 식 (10)에 의한 $score$가 최대인 단백질은 $PC_{cur}$에 추가되며, 이 단백질과 간선으로 직접 연결된 단백질들이 $Next_{cur}$에 추가된다. 또한, $score$가 최대인 단백질이 추가된 $PC_{cur}$에 대해서 식 (4)를 이용해서 $CC(PC_{cur})$를 계산하고, 식 (7)을 이용해서 $SCSM(PC_{cur})$을 계산한다.

위의 두 단계를 거친 후, 갱신된 $PC_{cur}$ 및 $Next_{cur}$에 대해서 단계 1과 2가 반복된다. 네트워크의 탐색은 탐색할 단백질이 더 이상 존재하지 않을 때까지, 즉 $Next_{cur}$가 비게 되는 경우 까지 반복된다. 이렇게 찾아진 $PC_{cur}$은 $R$에 추가된다.

네트워크의 탐색 과정에서, 현재까지 찾은 단백질의 집합인 $PC_{cur}$의 다음에 추가될 단백질의 집합을 의미하는 $Next_{cur}$은 이미 탐색된 단백질 복합체의 집합인 $R$에 포함되는 단백질을 포함할 수 있다. 즉, $R$의 각 단백질 복합체는 서로 간에 중복을 허용한다. 또한, 네트워크 탐색 과정에서 $PC_{cur}$에 추가되는 단백질은 $Next_{cur}$의 단백질 중 최대의 $score$를 가지는 것인데, 이것은 추가되는 단백질이 $SCSM$을 많이 유지하면서, 동시에 부분 네트워크를 서로 간에 강하게 연결하도록 하는 것임을 의미한다. 이상 설명된 네트워크 탐색 알고리즘은 그림 2와 같은 수도 코드로 정리할 수 있다.

```
전역 변수 : CC_thre, SCS_thre

Procedure FindProteinComplexes(E)    // 스코어링된 간선의 집합 E
    E 를 스코어에 대해 내림차순으로 정렬;
    R ← ∅;
    Examined_proteins ← ∅;
    for each e = (g_i, g_j) in E
        if(g_i ∉ Examined_proteins and g_j ∉ Examined_proteins)
            PC_cur ← {g_i, g_j};                                    // 초기화
            Next_cur ← {g_i및 g_j와 간선으로 직접 연결된 단백질의 집합};
            PC_cur.CC ← CC(PC_cur);                                 // = 1, 식 (4)
            PC_cur.SCS ← SCS1(PC_cur);                              // 식 (3)
            PC_cur ← traverseNetwork(PC_cur, Next_cur);
            Examined_proteins ← Examined_proteins ∪ PC_cur;
            R 에 PC_cur을 추가함;
        end
    end
    return R;

Procedure traverseNetwork(PC_cur, Next_cur)
    if(Next_cur is empty)
        return;
    else
        best_score ← 0;
        for each protein p in Next_cur
            PC_cur' ← PC_cur ∪ p;                                   // 단계 1
            PC_cur'.CC ← CC(PC_cur');                               // 식 (4)
            PC_cur'.SCS ← SCSM(PC_cur');                            // 식 (7)
            SCS_ratio ← |PC_cur'.SCS|/|PC_cur.SCS |;
            CC_ratio ← CC_cur'.SCS /CC_cur.SCS ;
            p.score ← SCS_ratio X CC_ratio;

            if(SCS_ratio < SCS_thre and CC_ratio < CC_thre)         // 단계 2
                remove p from Next_cur;
            else
                if(best_score < p.score)
                    best_score ← p.score;
                    best_p ← p;
                end
            end
        end

        PC_cur ← PC_cur ∪ best_p;
        Next_cur ← Next_cur ∪ {best_p와 간선으로 직접 연결된 단백질의 집합};
        PC_cur.CC ← CC(PC_cur);                                     // 식 (4)
        PC_cur.SCS ← SCSM(PC_cur);                                  // 식 (7)
        traverseNextork(PC_cur, Next_cur);
    end
```

그림 2 네트워크 탐색 알고리즘

## 3. 실험 결과

본 실험에서는 실제 PIN 및 유전자 발현 데이터을 이용해서 단백질 복합체를 탐색하고, 알려진 단백질 복합체 집합을 이용해서 그 정확도를 측정함으로써 2절에서 기술된 방법의 우수성을 보여준다.

### 3.1 실험 환경

본 실험에서 사용하는 PIN 데이터 및 유전자 mRNA 발현 데이터는 생물학적인 검증이 비교적 풍부하게 이루어진 Saccharomyces cerevisiae(yeast)에 대한 데이터이다. PIN 데이터는 2가지 종류로써, DIP[16]와 BioGRID[17] 데이터베이스로부터 수집했다. DIP 데이터셋은 검증이 완료된 데이터로, 그 신뢰도가 비교적 높은 대신 단백질의 수에 비해서 PPI의 수가 상대적으로 적다. 즉, 네트워크의 간선 밀도가 상대적으로 적다. 반면 BioGRID 데이터셋은 DIP 데이터셋에 비해 네트워크의 간선 밀도가 10배 가량 높고, 예측된 데이터가 많이 포함되어 있기 때문에, DIP 데이터셋에 비해 높은 거짓 긍정 오류율(false-positive rate)을 예상할 수 있다. 표 2는 수집한 네트워크 데이터의 버전과 크기를 보여주고 있다.

표 2 PIN 데이터셋 정보

| 데이터베이스 (버전) | 단백질의 개수 | PPI의 개수 |
|---|---|---|
| DIP (20071007) | 4,823 | 16,914 |
| BioGRID (3.1.69) | 5,920 | 162,378 |

또한, 유전자 mRNA 발현값 데이터는 [18]로부터 수집했다. 이 데이터는 2994개의 유전자의 mRNA 발현값을 173개의 조건에 대해서 측정했다. 마지막으로, 네트워크 탐색 알고리즘의 실행 결과를 검증하기 위해서 사용한 알려진 단백질 복합체(레퍼런스) 데이터셋 또한 2종류로써, MIPS[19] 및 CYC2008[20] 데이터베이스에서 수집할 수 있었다. 표 3은 레퍼런스 데이터셋의 정보를 보여준다.

표 3 레퍼런스 데이터셋의 정보

| 데이터베이스 | 단백질 복합체의 개수 | 단백질의 개수 | 단백질 복합체의 평균 단백질 개수 |
|---|---|---|---|
| MIPS | 81 | 885 | 12.35 |
| CYC2008 | 105 | 967 | 10.84 |

### 3.2 정확도의 측정 및 비교 실험

본 장에서는 네트워크 탐색 알고리즘의 정확도를 측정하고, 대표적인 네트워크 탐색 알고리즘인 MCODE[4], MCL[5], Ahn *et al.* [12]과의 비교 실험을 수행하였다. 이를 위해서 앞서 기술한 두 개의 PIN 데이터셋과 두 개의 레퍼런스 데이터셋을 조합해서 각 알고리즘 당 도합 4번의 비교 실험을 수행하였다. 레퍼런스 데이터셋 내의 어떤 레퍼런스 단백질 복합체가 각 알고리즘을 이용해서 찾은 단백질 복합체와 일치하는지의 여부는 affinity score를 이용해서 결정하였다. 레퍼런스 단백질 복합체의 단백질 집합을 $A$, 탐색한 단백질 복합체의 단백질 집합을 $B$라고 할 때, $A$와 $B$간의 affinity score인 $aff(A, B)$는 식 (11)과 같이 계산된다. 어떤 레퍼런스 단백질 복합체에 대해서, affinity score가 0.2 이상 [21]인 단백질 복합체의 탐색에 성공했을 경우 이 레퍼런스 단백질 복합체를 잘 찾아냈다고 판단했다.

$$aff(A,B)=\frac{|A\cap B|^2}{|A|\times|B|} \tag{11}$$

알고리즘을 이용해서 찾아낸 단백질 복합체 집합을 $C$, 레퍼런스 데이터셋의 단백질 복합체 집합을 $R$이라 할 때, 알고리즘의 성능은 recall, precision, F1 score로 평가할 수 있으며, 식 (12)와 같이 계산할 수 있다. 간단히 정리하면, recall은 레퍼런스 데이터셋을 얼마나 많이 찾았는지를, precision은 찾은 결과가 얼마나 레퍼런스 데이터셋과 일치하는지를 나타내며, F1 score는 전체적인 테스트 결과의 정확도를 의미한다.

$$Recall=\frac{|R_{hit}|}{|R|} \tag{12}$$

$$\mathrm{Pr}ecision=\frac{|C_{hit}|}{|C|}$$

$$F1\,score=\frac{2\times Recall\times \mathrm{Pr}ecision}{Recall+\mathrm{Pr}ecision}$$

$$, where\ C_{hit}=c_i\in C|aff(c_i,r_j)\geq 0.2,\ \exists r_j\in R$$

$$\text{and}\ R_{hit}=r_i\in R|aff(r_i,c_j)\geq 0.2,\ \exists c_j\in C$$

표 4는 검증 결과를 보여주고 있다. 각각의 알고리즘에 대하여는 파라미터를 바꾸어 가면서 최적의 결과를 도출하는 최적의 파라미터를 구했다. 본 논문에서의 알고리즘의 경우, 표 4의 최적 파라미터는 *CC_thre*값만을 의미하며, 나머지 사용자 입력값인 *SCS_thre*는 모든 실험에서 최적의 성능을 보여준 값인 0.6으로 고정했다.

표 4를 통해서 본 논문에서 제안하는 알고리즘이 전체적으로 높은 F1 score를 보이는 것을 확인할 수 있다. 특히, PPI의 수가 많은 BioGRID 데이터셋 같은 경우, 다른 알고리즘에 비해서 월등히 높은 F1 score를 보이고 있다. BioGRID 데이터셋은 상대적으로 오류율이 높을 것으로 예상되므로, BioGRID 데이터셋에 대해서 본 논문의 알고리즘이 효과적이라는 것으로부터 실제 유전자 발현값 데이터를 추가적으로 사용한 것이 단백질 복합체를 탐색하는 것에 효과적이라는 사실을 유추할 수 있다.

전체적으로 중복을 허용하는 방법인 본 논문의 알고

표 4 비교 실험 결과

| PIN 데이터셋 | 레퍼런스 데이터셋 | 알고리즘 | 최적 파라미터 | 단백질 복합체 개수 | Recall | Precision | F1 score |
|---|---|---|---|---|---|---|---|
| DIP | MIPS | Ours | *CC_thre* = 0.65 | 188 | 0.6173 | 0.2660 | 0.3717 |
| | | Ahn *et al.* | Partition_density = 0.27 | 299 | 0.7901 | 0.2140 | 0.3368 |
| | | MCL | Granularity = 2.00 | 253 | 0.5161 | 0.1304 | 0.2082 |
| | | MCODE | Node_score = 0.10 | 39 | 0.1774 | 0.2820 | 0.2178 |
| | CYC2008 | Ours | *CC_thre* = 0.65 | 188 | 0.6250 | 0.3191 | 0.4225 |
| | | Ahn *et al.* | Partition_density = 0.27 | 299 | 0.9479 | 0.3043 | 0.4608 |
| | | MCL | Granularity = 2.40 | 200 | 0.5063 | 0.2050 | 0.2918 |
| | | MCODE | Node_score = 0.10 | 39 | 0.2278 | 0.4871 | 0.3104 |
| BioGRID | MIPS | Ours | *CC_thre* = 0.70 | 682 | 1.0000 | 0.1554 | 0.2690 |
| | | Ahn *et al.* | Partition_density = 0.20 | 2581 | 1.0000 | 0.0593 | 0.1119 |
| | | MCL | Granularity = 3.60 | 53 | 0.0909 | 0.1320 | 0.1076 |
| | | MCODE | Node_score = 0.10 | 90 | 0.0649 | 0.0555 | 0.0598 |
| | CYC2008 | Ours | *CC_thre* = 0.70 | 682 | 1.0000 | 0.1892 | 0.3181 |
| | | Ahn *et al.* | Partition_density = 0.20 | 2581 | 1.0000 | 0.1015 | 0.1843 |
| | | MCL | Granularity = 3.00 | 55 | 0.0952 | 0.2000 | 0.1290 |
| | | MCODE | Node_score = 0.10 | 90 | 0.0476 | 0.0555 | 0.0512 |

리즘 및 Ahn *et al.*의 경우 그렇지 않은 방법인 MCL과 MCODE에 비해서 많은 수의 단백질 복합체를 찾아내는 것을 확인할 수 있다. 이런 현상은 간선의 수가 부족한 DIP 데이터셋보다 간선의 수가 많은 BioGRID 데이터셋에서 뚜렷이 나타난다. 찾아낸 단백질 복합체의 수가 많을수록 레퍼런스 단백질 복합체를 보다 많이 찾아낼 수 있으므로 recall은 상대적으로 높아진다.

단백질 복합체의 생물학적 검증은 높은 비용을 필요로 한다. 따라서, 단백질 복합체의 예측 알고리즘은 높은 precision으로 단백질 복합체를 예측해주는 것이 바람직하다. 이러한 측면에서, 본 논문의 알고리즘은 생물학적 검증 비용을 상당 부분 낮추어 줄 수 있다.

보통 많은 수의 단백질 복합체를 찾아낼수록 recall은 높아지고, precision은 낮아지는 경향을 보인다. 하지만 BioGrid 데이터셋과 MIPS 레퍼런스 데이터셋을 이용한 실험을 이용한 경우, 본 논문의 알고리즘은 MCL보다 12배 많은 단백질 복합체를 찾아내었음에도 불구하고, 1.2배 높은 precision을 보여주고 있으며, MCODE의 경우에는 7배 많은 단백질 복합체를 찾아내었음에도 불구하고, 2.8배 높은 precision을 보여주고 있다. 이는 본 논문의 알고리즘이 기존에 단백질 복합체를 예측하기 위해서 많이 사용되어왔던 방법보다 효율적으로 단백질 복합체를 찾아줄 수 있음을 보여주고 있다.

## 4. 결 론

본 논문에서는 PIN에서 단백질 복합체를 찾아내는 네트워크 클러스터링 알고리즘을 제안한다. 이 알고리즘은 높은 오류율을 보이는 PIN의 한계를 극복하기 위해서 유전자 발현값 데이터를 추가적으로 이용한다. 또한 하나의 단백질이 여러 단백질 복합체에 속할 수 있다는 사실을 반영해서 중복을 허용하는 네트워크 탐색 방법을 도입했다. 결과적으로 본 논문에서 제안하는 알고리즘은 기존의 네트워크 클러스터링 알고리즘에 비해서 전체적으로 높은 정확도를 보임을 확인할 수 있었다.

향후 연구로, 본 논문에서 제안하는 알고리즘을 클러스터 간의 계층적 관계를 보여줄 수 있도록 발전시킬 예정이다. 이 방법을 통해서 탐색된 단백질 복합체 간의 계층적 구조를 밝히고, Gene Ontology 데이터베이스 등을 통해서 단백질 복합체의 기능을 유추하는 연구를 계획하고 있다.

## 참 고 문 헌

[1] A. Kumar and M. Snyder, "Protein complexes take the bait," *Nature*, vol.415, pp.123-124, 2002.

[2] S. Fields and O. Song, "A novel genetic system to detect protein-protein interactions," *Nature*, vol.340, pp.245-245, 1989.

[3] Y. Ho, A. Gruhler *et al.*, "Systematic identification of protein complexes in Saccharomyces cerevisiae by mass spectrometry," *Nature*, vol.415, pp.180-183, 2002.

[4] G. D. Bader and C. W. Hogue, "An automated method for finding molecular complexes in large protein interaction networks," *BMC Bioinformatics,* vol.4, no.2, 2003.

[5] A. J. Enright, S. Van Dongen and C. A. Ouzounis, "An efficient algorithm for large-scale detection of protein families," *Nucleic Acids Research*, vol.30, no.7, pp.1575-1584, 2002.

[6] S. Brohee and J. van Helden, "Evaluation of clustering algorithms for protein-protein interaction networks," *BMC Bioinformatics,* vol.7, no.488, 2006.

[7] J. Vlasblom and S. Wodak, "Markov clustering versus affinity propagation for the partitioning of protein interaction graphs," *BMC bioinformatics,* vol.10, no.99, 2009.
[8] M. Li, J. Chen, J. Wang, B. Hu, and G. Chen, "Modifying the DPClus algorithm for identifying protein complexes based on new topological structures," *BMC Bioinformatics,* vol.9, no.398, 2008.
[9] M. Altaf-Ul-Amin, Y. Shinbo, K. Mihara, K. Kurokawa and S. Kanaya, "Development and implementation of an algorithm for detection of protein complexes in large interaction networks," *BMC Bioinformatics,* vol.7, no.207, 2006.
[10] B. Adamcsek, G. Palla, I. Farkas, I. Derenyi and T. Vicsek, "CFinder:locating cliques and overlapping modules in biological networks," *Bioinformatics*, vol.22, no.8, pp.1021-1023, 2006.
[11] G. Palla, I. Derenyi, I. Farkas and T. Vicsek "Uncovering the overlapping community structure of complex networks in nature and society," *Nature*, vol.435, pp.814-818, 2005.
[12] Y. Ahn, J. P. Bagrow and S. Lehmann, "Link communities reveal multiscale complexity in networks," *Nature*, vol.466, pp.761-765, 2010.
[13] C. M. Deane, L. Salwinski, I. Xenarios and D. Eisenberg, "Protein interactions: Two methods for assessment of the reliability of high throughput observations," *Molecular Cell Proteomics*, vol.1, pp.349-356, 2002.
[14] T. Ito, K. Ota, H. Kubota, Y. Yamaguchi, T. Chiba, K. Sakuraba and M. Yosida, "Roles for the two-hybrid system in exploration of the yeast protein interactome," *Molecular Cell Proteomics*, vol.1, pp.561-566, 2002.
[15] J. Ahn, Y. Yoon and S. Park, "Noise-robust algorithm for identifying functionally associated biclusters from gene expression data," *Information Sciences*, vol.181, pp.435-449, 2011.
[16] L. Salwinski, C. S. Miller, A. J. Smith, F. K. Pettit, J. U. Bowie, and D. Eisenberg, "The database of interacting proteins:2004 update," *Nucleic Acids Research*, vol.32, no. Database issue, pp. D449-D451, 2004.
[17] C. Stark, B. J. Breitkreutz, T. Reguly, L. Boucher, A. Breitkreutz, and M. Tyers, "BioGRID: a. general repository for interaction datasets," *Nucleic Acids Research*, vol.34, no. Database issue, pp. D535-D539, 2006.
[18] A. P. Gasch, P. T. Spellman, C. M. Kao, O. Carmel-Harel, M. B. Eisen, G. Storz, D. Bostein and P. O. Brown, "Genomic expression programs in the response of yeast cells to environmental changes," *Molecular Biology of the Cell*, vol.11, no.12, pp.4241-4257, 2000.
[19] U. Güldener *et al.*, "CYGD: the comprehensive yeast genome database," *Nucleic Acids Research*, vol.33, no. Database issue, pp.D364-D368, 2005.
[20] S. Pu, J. Wong, B. Turner, E. Cho and S. Wodak, "Up-to-date catalogues of yeast protein complexes," *Nucleic acids research*, vol.37, no.3, pp.825-831, 2009.
[21] Y. Yeu, J. Ahn, Y. Yoon and S. Park, "Protein Complex Discovery from Protein Interaction Network with High False-Positive Rate," *Poster abstracts of 9th European Conference on Evolutionary Computation, Machine Learning and Data Mining in Bioinformatics* (*EvoBIO'11*), pp. 177-182, 2011.

안 재 균
2006년 연세대학교 컴퓨터과학과 졸업(학사). 2009년 연세대학교 컴퓨터과학과 졸업(석사). 2010년~현재 연세대학교 컴퓨터과학과 박사과정. 관심분야는 바이오인포메틱스, 데이터 마이닝

여 윤 구
2009년 연세대학교 컴퓨터과학과 졸업(학사). 2011년 연세대학교 컴퓨터과학과 졸업(석사). 관심분야는 데이터베이스, 데이터 마이닝, 바이오인포매틱스

윤 영 미
1981년 서울대학교 자연과학대학 졸업(학사). 1983년 오하이오 주립대학 수학과(학사수료). 1987년 스탠포드대학교 컴퓨터과학과 졸업(이학석사). 2008년 연세대학교 컴퓨터과학과 졸업(공학박사) 1987년~1993년 IntelliGenetics Inc., California, USA, Software Engineer. 1995년~현재 가천대학교 교수, 컴퓨터공학과. 관심분야는 데이터베이스 시스템, 데이터 마이닝, 바이오인포매틱스

박 상 현
1989년 서울대학교 컴퓨터공학과(공학사). 1991년 서울대학교 컴퓨터공학과(공학석사). 2001년 UCLA대학교 전산학과(공학박사). 1991년~1996년 대우통신 연구원. 2001년~2002년 IBM T. J. Watson Research Center Post-Doctoral Fellow. 2002년~2003년 포항공과대학교 컴퓨터공학과 조교수. 2003년~2006년 연세대학교 컴퓨터과학과 조교수. 2006년~2011년 연세대학교 컴퓨터과학과 부교수. 2011년~현재 연세대학교 컴퓨터과학과 정교수. 관심분야는 데이터베이스, 데이터 마이닝, 바이오인포매틱스, 적응적 저장장치 시스템