DOI: 10.3745/KIPSTD.2009.16-D.3.327

유전자 발현 데이터에 적용한 거시적인 바이클러스터링 기법

안 재 균[†] · 윤 영 미[††] · 박 상 현[†††]

요 약

마이크로어레이 데이터는 유전자의 집합이 어떠한 조건 혹은 샘플의 집합 하에서 얼마나 발현되는지를 수치화한 2차원 행렬 데이터이다. 바이클러스터는 마이크로어레이의 샘플의 부분 집합과 이 샘플 부분 집합 하에서 일정한 증감 패턴을 보이는 유전자의 부분 집합을 말한다. 이렇게 같은 패턴을 보이는 유전자의 부분 집합은 일정한 정도의 유의 수준으로 비슷한 기능을 한다고 말할 수 있다. 따라서 바이클러스터링 알고리즘은 같은 기능에 연관된 유전자의 집합과, 이 기능이 발현되고 있는 조건의 집합을 밝혀내는데 있어서 매우 유용하다. 본 논문에서는 다항식 시간 복잡도를 유지하면서, 높은 기능적 상관관계를 가지는 바이클러스터를 밝혀 낼 수 있는 알고리즘을 제안한다. 이 알고리즘은 1) 마이크로어레이 데이터에 심한 노이즈가 있을 경우 패턴으로 인식하지 못하는 기존 알고리즘과 달리, 노이즈 레벨이 심하더라도 거시적으로 비슷한 모양을 보이는 패턴을 찾아내는 방식을 이용하여 숨어있는 패턴들을 찾아낼 수 있고, 2) 바이클러스터 상호간에 오버랩을 허용하며, 또한 다양성이 보장되는 복수의 바이클러스터를 찾아내며, 3) 찾아진 유전자 부분 집합의 기능적 상관관계가 매우 높은 특성을 지니고, 4) 유전자 및 샘플의 순서와 상관없이 결정적인(deterministic) 결과를 도출한다. 또한 본 논문에서는 알고리즘이 찾아낸 바이클러스터의 기능적 상관관계의 정도와, 비교 알고리즘이 찾아낸 바이클러스터의 기능적 상관관계의 정도를 유전자 온톨로지(Gene Ontology)를 통해서 측정함으로써 비교하고 있다.

키워드 : 데이터 마이닝, 바이클러스터링, 유전자 표현형 데이터 분석, 마이크로어레이 데이터 분석, 노이즈

Macroscopic Biclustering of Gene Expression Data

Jaegyoon Ahn[†] · Youngmi Yoon[††] · Sanghyun Park[†††]

ABSTRACT

A microarray dataset is 2-dimensional dataset with a set of genes and a set of conditions. A bicluster is a subset of genes that show similar behavior within a subset of conditions. Genes that show similar behavior can be considered to have same cellular functions. Thus, biclustering algorithm is a useful tool to uncover groups of genes involved in the same cellular process and groups of conditions which take place in this process. We are proposing a polynomial time algorithm to identify functionally highly correlated biclusters. Our algorithm identifies 1) the gene set that has hidden patterns even if the level of noise is high, 2) the multiple, possibly overlapped, and diverse gene sets, 3) gene sets whose functional association is strongly high, and 4) deterministic biclustering results. We validated the level of functional association of our method, and compared with current methods using GO.

Keywords : Data Mining, Biclustering, Gene Expression Data Analysis, Microarray Analysis, Noise

1. 서 론

마이크로어레이 데이터는 유전자의 집합이 어떠한 조건 혹은 샘플의 집합 하에서 얼마나 발현되는지를 수치화한 2차원 행렬 데이터이며, 그 예는 <표 1>과 같다. 마이크로어레이 분석의 목적 중 하나는 마이크로어레이 에 참여하는 유전자의 기능을 밝히는 것이다. 이를 위해서 기존의 많은 분석 방법은 마이크로어레이의 모든 조건 하에서 유전자의 발현값을 조사함으로써, 기능적 상관관계를 가지는 유전자를 클러스터링하는 방법을 취했다[1, 9].

그러나 모든 조건에서 특정 기능과 관련된 유전자의 발현 정도를 관찰할 수 있는 것은 아니다. 즉, 우리는 어떤 유전자의 부분 집합이 특정한 실험적 조건 집합 하에서 상관관계를 가진다고 기대할 수 있지만, 다른 조건에서는 이 유전자의 부분 집합의 상관관계는 사라질 수 있다[2]. 서로 간에 상관관계가 있는 유전자의 부분 집합을 밝혀내는 것은 유전자 집합의 기능을 밝히고, 나아가 유전자 제어 네트워크를 밝히는 중요한 역할을 할 수 있다. Cheng [3]은 마이크로어레이 행렬 데이터에서 서로 간에 밀접한 상관관계를 가지는 유전자의 집합과 샘플의 집합으로서 구성되는 부분 행렬을

※ 본 연구는 한국과학재단의 생물정보학연구개발사업(2008-2004103)의 지원을 받아 수행되었습니다.
† 준 회 원 : 연세대학교 컴퓨터과학과 석사과정
†† 종신회원 : 가천의과학대학교 부교수
††† 종신회원 : 연세대학교 컴퓨터과학과 부교수(교신저자)
논문접수 : 2009년 1월 29일
수 정 일 : 1차 2009년 4월 1일
심사완료 : 2009년 4월 1일

찾는 데이터 마이닝 기법을 바이클러스터링이라고 명명했고, 이러한 부분 행렬을 바이클러스터라고 명명했다.

바이클러스터링 프로세스는 NP-Hard임이 증명되었고 [3], 지금까지 제시된 많은 바이클러스터링 알고리즘은 모두 휴리스틱한 방법이나 확률 통계적 접근 방식을 취했다. 따라서 각 알고리즘의 장단점은 모두 다르고 각 알고리즘이 밝혀낼 수 있는 패턴 또한 다양하다. 바이클러스터링 알고리즘이 밝혀낼 수 있는 패턴의 종류에 따라서 이들을 크게 두 그룹으로 나눌 수 있다.

1.1 가법(additive) 패턴 혹은 승법(multiplicative) 패턴

바이클러스터는 (그림 1)[5]과 같이 유전자의 그래프로 표시할 수 있다. (그림 1)의 각 그래프에서 유전자는 꺾인 선으로, 샘플은 a~j로 표기되어 있다. (그림 1-(a))는 유전자 간에 패턴을 보이지 않는다. (그림1-(b))는 샘플 "bchje"에서 유전자간의 가법(additive) 패턴을, (그림 1-(c))는 샘플 "fdagi"에서 승법(multiplicative) 패턴을 보여준다.

δ-biclustering[3]은 바이클러스터를 찾기 위해서 마이크로어레이 데이터의 부분 행렬의 최소제곱잔여(mean squared residue)를 계산한다. 행렬의 최소제곱잔여는 행렬의 각 행 및 열끼리 어느 정도의 상관관계를 가지고 있는가를 나타내며, 이 값이 작을수록 서로 간에 큰 상관관계를 가진다. δ-biclustering은 이 최소제곱잔여가 작은 부분 행렬을 찾아내며, 결과적으로 이러한 부분 행렬은 가법 패턴이나 승법 패턴을 보이는 유전자 및 샘플로 이루어진다. 이 알고리즘의 단점은 노이즈를 많이 허용하지 않는다는 것이다. 그러므로 이 알고리즘은 엄격한 패턴만을 찾는다고 할 수 있다. 또한 이 알고리즘은 찾은 바이클러스터의 부분 행렬을 임의의 실수로 채워 넣음으로써 첫 번째 바이클러스터 이후로 찾아지는 바이클러스터의 신뢰도를 떨어뜨리며, 바이클러스터간의 오버래핑도 허용하지 않는다.

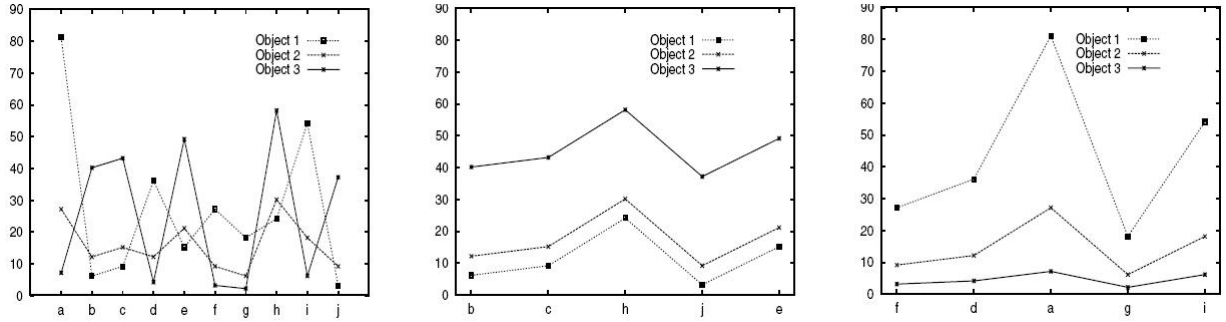
P-bicluster[4] 또한 최소제곱잔여가 최소가 되는 부분 행렬을 찾는 알고리즘이다. P-bicluster는 2×2 행렬로부터 시작해서 점차적으로 행과 열을 늘려가는 방법을 취함으로써 δ-biclustering보다 빠른 성능을 보이고 있다. 하지만 δ-biclustering의 많은 문제점들은 해결하고 있지 못하다.

p-Cluster[5]는 모든 열 쌍과 행 쌍에 대해서 Maximum Dimension Set(MDS)이라 불리는 최대(maximal) 클러스터를 찾아낸다. 그 후 열 쌍 MDS와 행 쌍 MDS를 가지치기(pruning)하고 프리픽스 트리(prefix tree) 구조에 대입시켜 바이클러스터를 구해나간다. 이 알고리즘 또한 큰 노이즈를 허용하지 않는다는 단점이 있으며, 알고리즘의 시간 복잡도가 샘플의 개수에 대해서 지수적으로 증가한다.

Tri-Cluster[6]는 차원 마이크로어레이 데이터에 대해서 마이닝을 시도한 첫 번째 알고리즘으로, 사용자 입력값 ε내로 비슷하게 오르내리는 유전자 집합을 노드(node)로 하는 깊이 우선 트리(depth first tree)를 만들어서 바이클러스터를 찾아낸다. 만약 ε이 너무 크다면 깊이 우선 트리가 너무 커지게 되고, 너무 작다면 허용 노이즈가 불충분하다는 문제점을 가진다. 그리고 이 알고리즘의 시간 복잡도는 ε 및 샘플의 개수에 대해서 지수적으로 증가한다.

reg-Cluster[7]는 가법패턴과 승법 패턴을 동시에 찾는다. 이 알고리즘은 d_{ij} 를 조건 c_i와 c_j 사이의 유전자 발현값의 차이로 정의하고 d_{01}과 d_{ij} 사이의 유전자 발현값의 비율이 ε 이내인 유전자들을 찾아낸다. 이 알고리즘이 기존의 알고리즘과 차별되는 점은 기존의 것들이 가법 패턴과 승법 패턴을 독립적으로 찾는데 비해서 이 알고리즘은 가법 패턴과 승법 패턴을 동시에 찾는다는 것이다. 하지만 이 알고리즘은 Tri-Cluster와 같은 깊이 우선 트리를 형성하므로 Tri-Cluster와 비슷한 문제를 가지게 된다. 깊이 우선 트리는 레벨이 늘어남에 따라서 지수적으로 커지므로, 깊이 우선 트리를 만드는 알고리즘은 공통적으로 생성된 바이클러스터가 매우 많다는 문제점을 가진다. 또한 이렇게 많은 바이클러스터들은 대부분 서로 약간씩만 다르다는 특징을 가지며, 이 많은 바이클러스터를 모두 검사해서 대표성을 가지는 몇 개만 선택하는 것은 매우 어려운 문제이다.

Liu[8]은 서로 간에 가법패턴을 보이는 유전자 집합을 그리디 방법(greedy method)으로 골라냄으로써 다항식 시간 복잡도 (polynomial time complexity)를 가지는 알고리즘을 제안했다. 따라서 [8]는 reg-Cluster 및 Tri-Cluster가 가지는 문제를 해결하고 있으나, 그리디 알고리즘의 근본적인 문제점인 최적해를 보장할 수 없다는 문제는 여전히 해결하지 못하고 있으며, 또한 reg-Cluster 및 Tri-Cluster에 비해서 다양한 바이클러스터를 만들지 못한다는 문제를 추가로 가진다.



(a) 기본형태 (b) 가법(additive) 패턴 (c) 승법(multiplicative) 패턴

(그림 1) 유전자 오브젝트와 샘플

1.2 정렬된 시퀀스에 의한 패턴

OPSM[2]은 각 행에 대해서 유전자 발현값을 정렬한 후 각 열에 대해서 적당한 연산을 적용한 마이크로어레이 행렬의 부분 행렬로서 바이클러스터를 정의한다. 비록 OPSM은 매우 좋은 GO검증 결과를 보여주고 있으나[9], OPSM이 한 번 실행할 때마다 하나씩 밖에 결과를 찾아내지 못한다는 것은 문제점이다. OP-Cluster[10]와 KiWi[11]는 OPSM과 부분적으로 다른 알고리즘을 사용하나, 기본적으로 바이클러스터의 정의는 OPSM과 같다. 공통적으로 OPSM 기반 알고리즘들은, 순서가 지켜지지 않으면서 생물학적으로 의미 있는 패턴을 놓칠 위험을 가지고 있다[12].

상기 기술된 기존의 바이클러스터링 기법들은 바이클러스터 내의 인접한 샘플과 샘플, 혹은 유전자와 유전자 사이의 발현값의 차이 혹은 순서로써 패턴이 성립되는가의 여부를 판단했다. 그러므로 마이크로 어레이 데이터의 노이즈가 높을 경우, 미시적인 패턴이 성립되지 않는 많은 유전자를 놓칠 가능성이 있다. 본 논문에서는 높은 기능적 상관관계를 가지는 유전자 집합을 가지는 바이클러스터를 도출하기 위한 새로운 바이클러스터링 알고리즘인 Macro-Cluster를 제안한다. Macro-Cluster는 1) 거시적인 관점에서 봤을 때 비슷한 증감 패턴을 가지는 유전자의 집합을 클러스터링하는 방법을 통해서 이러한 단점을 해결한다. 구체적으로, Macro-Cluster는 비닝(Binning)[20]을 통해서 큰 노이즈를 허용한다. 그러므로 허용 노이즈 정도가 커도 Macro-Cluster의 시간 및 공간 복잡도는 지수적으로 증가하지 않는다. 즉, Macro-Cluster는 노이즈 레벨에 대해 견고한 특성을 가진다. 2) 다수의 오버래핑 가능한 유전자 집합을 찾아내는 동시에, 찾아진 유전자 집합은 그 다양성이 보장된다. 3) 기능적 상관관계의 정도가 매우 높은 바이클러스터를 잘 찾아낸다. 이 기능적 상관관계의 정도는 GO 데이터베이스를 이용해서 검증된다. 4) 마이크로어레이의 샘플 및 유전자의 순서와 상관없이 같은 결과를 도출한다. 즉, Macro-Cluster는 결정적(deterministic)이라고 할 수 있다.

마이크로어레이 데이터를 이용해서 유전자 발현 네트워크 혹은 단백질 네트워크를 구성하는 작업은 기능 유전학(Functional Genomics) 등의 응용 연구에 중요한 역할을 담당한다. 이러한 작업에서 비슷한 기능을 하는 유전자 집합을 찾는 것은 기본적이고 필수적인 일이다. 이미 언급한 Macro-Cluster가 가지는 네 가지 특징은 Macro-Cluster가 효과적으로 다양한 크기의 다양한 유전자 집합을 찾을 수 있음을 설명하고 있으며, Macro-Cluster가 현재 마이크로어레이 분석 시 쓰이고 있는 클러스터링 알고리즘들을 대체할 경우 더욱 우수한 유전자 발현 네트워크 혹은 단백질 네트워크를 구성할 수 있음을 기대할 수 있다.

본 논문의 구성은 다음과 같다. 2장에서는 본 논문에서 제안하는 Macro-Cluster를 정의하고 세부적인 모델을 설명한다. 3장에서는 2장에서 정의된 Macro-Cluster를 마이닝할 수 있는 알고리즘을 설명한다. 4장에서는 실험을 통하여 제안 모델 및 알고리즘의 성능을 평가한다. 마지막으로 5장에서는 본 논문을 요약하고 앞으로의 연구 방향을 제시한다.

2. Macro-Cluster의 기호, 모델 및 정의

이 장은 알고리즘을 기술하기에 앞서 설명해야 할 기호, 모델 및 정의에 대해 서술한다.

2.1 기호

G	마이크로어레이 데이터의 전체 유전자 집합, 전체 m개의 유전자가 있음
S	마이크로어레이 데이터의 전체 샘플 집합, 전체 n개의 샘플이 있음
(O, T)	O ⊆ G, T ⊆ S를 만족하는 마이크로어레이 데이터의 부분 행렬 데이터
$g_0, g_1, \cdots$	O의 유전자들
$s_0, s_1, \cdots$	T의 샘플들
c_{ij}	유전자 g_i의 샘플 s_j에서의 발현값
d	범위 계산에 필요한 범위 증가값 (> 1, 사용자 입력값)
a	범위 계산에 필요한 초기값 (> 0, 사용자 입력값)
mg	Macro-Cluster를 만족하는 최소 유전자의 수(≥ 2, 사용자 입력값)
ms	Macro-Cluster를 만족하는 최소 샘플의 수(≥ 2, 사용자 입력값)
rt	두 Macro-Cluster가 유사함을 판단할 수 있는 척도(사용자 입력값)

2.2 모델과 정의

마이크로어레이 데이터는 각 열 데이터의 집합이다. 즉, 2차원 마이크로어레이 데이터의 각 열은 하나의 마이크로어레이 실험을 뜻한다. 예를 들어, <표 1>의 마이크로어레이의 제1열은 샘플 s_0에서 10개의 유전자에 대한 실험을 말한다. 따라서 각 열마다 실험적인 편차가 있을 수 있다. 그러므로 이러한 실험적 편차를 없애기 위해서 마이크로어레이의 각 샘플마다 전체 유전자 집합 G를 정규화한다. 정규화 시 각 발현값 x_i는 다음의 식에 의해 z_i로 변경된다.

$$z_i = \frac{x_i - \bar{x}}{\sigma}, x_i : \text{유전자발현값}, \bar{x} : \text{평균}, \sigma : \text{표준편차}$$

<표 1>의 10 × 5 마이크로어레이 데이터에서, 각 칸은 유전자의 발현값(괄호 안의 값) 및 그 값을 정규화한 값을 나타낸다. 예를 들어 유전자 YAL003W의 샘플 s0 하에서의 발현값은 0.15이고, 정규화된 유전자 발현값은 0.916794가 된다.

그 후 각 샘플마다 정규화된 발현값이 일정한 범위 내에 있는 유전자들을 클러스터링한다. 만약 샘플 집합 T에 속하는 각 샘플에서의 클러스터링된 유전자의 집합 이 공통된 유전자 집합 O를 가진다면, O는 이러한 T의 각 샘플에 대해서 비슷한 성향을 가진다고 할 수 있다. 이때, $C = (O, T)$를 Macro-Cluster라고 한다. Macro -Cluster의 유전자 집합 O의 유전자 각각이 실제 높은 확률로 같은 기능을 한다는 사실은 4장의 실험에서 실제 마이크로어레이 데이터와 이미 구축된 유전자 기능 데이터베이스를 이용해서 증명하고 있다.

〈표 1〉 10 x 5 마이크로어레이 데이터 및 정규화 예제

gene \ sample	s_0	s_1	s_2	s_3	s_4
YAL003W (g_0)	0.9168 (0.15)	-0.2817(-0.07)	-1.2823(-0.25)	-0.9802 (-0.3)	-2.4850(-1.12)
YAL012W (g_1)	1.2462 (0.21)	0.0148 (0.03)	-0.2439 (0.18)	-0.9135(-0.27)	-0.4585(-0.32)
YAL014C (g_2)	-0.0714(-0.03)	-0.2817(-0.07)	-0.0024 (0.28)	0.3978 (0.32)	-0.3318(-0.27)
YAL015C (g_3)	-1.2791(-0.25)	1.6459 (0.58)	1.1808 (0.77)	0.3089 (0.28)	1.1626 (0.32)
YAL016W (g_4)	0.6972 (0.11)	0.0445 (0.04)	1.1325 (0.75)	1.5092 (0.82)	0.8840 (0.21)
YAL017W (g_5)	1.4109 (0.24)	0.8452 (0.31)	1.6155 (0.95)	-0.0467 (0.12)	0.8080 (0.18)
YAL021C (g_6)	-1.5536 (-0.3)	0.5783 (0.22)	-0.6303 (0.02)	-1.7360(-0.64)	0.5041 (0.06)
YAL022C (g_7)	-0.7301(-0.15)	-0.8155(-0.25)	-0.2439 (0.18)	-0.1800 (0.06)	-0.0279(-0.15)
YAL029C (g_8)	0.0933 (0.00)	-2.2687(-0.74)	-1.5962(-0.38)	1.6203 (0.87)	-0.5091(-0.34)
YAL030W (g_9)	-0.7301(-0.15)	0.5190 (0.20)	0.0700 (0.31)	0.0200 (0.15)	0.4534 (0.04)

이상이 Macro-Cluster의 전체적인 모델이며, 이를 수학적으로 정의하기 위해서 먼저 유전자가 같이 클러스터링될 수 있는 범위에 대해서 정의한다.

[정의 1] (R_i). R_i 는 [min_i, max_i)로 정의되는 범위이며, i가 임의의 정수이고, 사용자 입력값 a와 d가 임의의 양의 실수일 때, a와 d에 대해서 min_i와 max_i는 다음과 같이 계산된다.

1. i > 0 : $min_i = a \times (i-1)^d$, $max_i = a \times (i+1)^d$
2. i = 0 : $min_i = -a$, $max_i = a$
3. i < 0 : $min_i = -a \times (1-i)^d$, $max_i = -a \times (-1-i)^d$

범위 R_i가 (그림 2)에 설명된다. (그림 2)를 보면, 모든 범위가 일정 정도 겹치고 있기 때문에 모든 정규화된 유전자 발현값은 빠짐없이 하나 혹은 두 개의 범위에 속할 수 있음을 확인할 수 있고, 그 중 대부분은 2개의 범위에 속함을 알 수 있다. 만약 범위가 겹치지 않는다면, 즉 (그림 2)의 범위 R_i의 i가 모두 짝수이거나 혹은 모두 홀수라면, 범위의 경계에 있는 정규화된 유전자 발현값들이 서로 다른 범위로 흩어질 가능성이 높다. 예를 들어, 사용자 입력값 a와 d가 각각 0.1와 2.0일 때, R_0, R_2는 각각 [-0.1, 0.1)과 [0.1, 0.9)이다. 만약 R_1이 없다고 한다면, 정규화된 유전자 발현값 0.099는 R_0에, 0.101는 R_2에 속하게 되나, 실제로 두 발현값은 같은 범위에 들어가야 할 정도로 차이가 미미하다. 만약 [0, 0.4)인 R1이 있다면 이 두 발현값은 R_1에 속하게 되므로 이러한 위험이 없어진다.

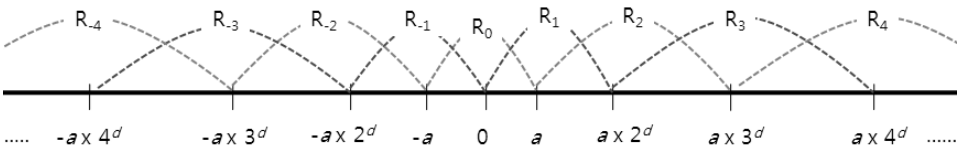
여기서 말하는 범위의 의미는, 임의의 두 유전자의 정규화된 발현값이 한 범위에 속한다면 두 유전자의 발현값은 같다고 할 수 있다는 것이다. 범위 R_i에 대해서 |i|가 커질수록 범위의 크기는 더 커진다. 그것은, 유전자들의 정규화된 발현값의 절대값이 작아질수록 서로 간에 다른 값이라고 판단할 수 있는 차이가 작고, 커질수록 서로 간에 다른 값이라고 판단할 수 있는 차이가 커지기 때문이다. 예를 들어서, 두 유전자의 정규화된 발현값이 각각 0.1과 0.2인 경우와 각각 2과 2.1인 경우를 비교해 보면, 두 경우 모두 차이는 0.1이지만 전자의 경우 후자의 경우보다 서로 다른 값이라고 판단하기 쉽다. 즉 0.1과 0.2를 다르게 판단하는 경우라도 2과 2.1은 같다고 판단할 수 있다는 것이다. 따라서 범위 R_i에 대해서 |i|가 커질수록 범위의 크기는 더 커져야 하며, 본 논문에서는 정의 1과 같이 범위를 정함으로써 |i|의 증가에 따라서 범위를 크게 한다.

[정의 2] (Macro-Cluster). $C = (O, T)$가 다음의 조건을 만족할 때 C를 Macro-Cluster라고 한다.

1. 임의의 범위 R에 대해서, T의 샘플 하에서 O의 모든 유전자의 발현값이 같은 R에 포함됨
2. $|O| \geq mg$ (≥ 2), $|T| \geq ms$ (≥ 2)

사용자 입력값 a와 d를 각각 0.1와 2.0으로 했을 때, 정의 1에 의해서 계산되는 각 범위와, 이 때 각 샘플 하에서 정규화된 유전자들이 계산된 각 범위에 binning된 결과가 <표 2>에 나와 있다. <표 2>에서, 유전자 g_6과 g_9는 샘플 s_0, s_1, s_2, s_4 하에서 같은 성향을 보임을 확인할 수 있다. 정의 2에 따라서 C = ({g_6, g_9}, {s_0, s_1, s_2, s_4})는 Macro-Cluster이다.

[정의 3] (p-MAC). Macro-Cluster $C = (O, T)$의 $|T|$ = p일 때, C를 p-MAC이라고 한다. 즉, p는 C의 샘플 수를 의미한다. 예를 들어, 샘플의 수 $|T|$ = 4일 때, C는 4-MAC이다. 정의 2의 2번에서 $ms \geq 2$이므로, 1-MAC은 존재할 수



(그림 2) 각 범위 R_i

〈표 2〉 a = 0.1, d = 2.0 으로 계산된 범위 및 범위에 따라 〈표 1〉의 유전자를 비닝(binning)한 결과

	s_0	s_1	s_2	s_3	s_4
R_{-4} : [-2.5, -0.9)	g_3, g_6	g_8	g_0	g_0, g_1, g_6	g_0
R_{-3} : [-1.6, -0.4)	g_3, **g_6**, g_7, **g_9**	g_7	g_0	g_0, g_1	g_1, g_8
R_{-2} : [-0.9, -0.1)	g_7, g_9	g_0, g_2, g_7	g_1, g_8	g_7	g_1, g_2, g_8
R_{-1} : [-0.4, 0)	g_2	g_0, g_2	g_1, g_2, g_8	g_5, g_7	g_2, g_7
R_0 : [-0.1, 0.1)	g_2, g_8	g_1, g_4	g_2, g_6, g_7	g_5, g_9	g_7
R_1 : [0, 0.4)	g_8	g_1, g_4	**g_6**, g_7, **g_9**	g_2, g_3, g_9	g_6
R_2 : [0.1, 0.9)	g_4	g_5, **g_6**, **g_9**	g_9	g_2, g_3	g_4, g_5, **g_6**, **g_9**
R_3 : [0.4, 1.6)	g_0, g_1, g_4, g_5	g_3, g_5, **g_6**, **g_9**	g_3, g_4	g_4	g_3, g_4, g_5, g_9
R_4 : [0.9, 2.5)	g_0, g_1, g_5	g_3	g_3, g_4, g_5	g_4, g_8	g_3

없으나, 예외적으로 유전자 집합과 샘플 한 개로 이루어진 1-MAC을 정의한다. 1-MAC은 binning된 각 유전자 집합과 그 때의 샘플로 이루어진다. 예를 들어서, <표 2>에서 도출될 수 있는 1-MAC은 ({g_3, g_6}, {s_0}), ({g_3, g_6, g_7, g_9}, {s_0}), ({g_7, g_9}, {s_0}), ({g_2}, {s_0}), ..., ({g_0, g_2, g_7}, {s_1}), ... 등등이 된다.

[정의 4] (p-MAC간의 유사도) 두 p-MAC $C_1 = (O_1, T_1)$과 $C_2 = (O_2, T_2)$이 있다고 하면, 유전자 집합 $O = O_1 \cap O_2$일 때, C_1 와 C_2 사이의 유사도는 $\max(|O|/|O_1|, |O|/|O_2|)$라고 표현되며, 이때 max(a, b) 는 a ≥ b 일 때, a를 반환하는 함수이다. 우리는 $\max(|O|/|O_1|, |O|/|O_2|) \geq rt$일 때, 두 p-MAC이 유사하다고 말한다. 여기서 유전자 집합의 유사도만으로 두 p-MAC의 유사성을 측정할 수 있는 이유는, 샘플 집합이 아닌 유전자 집합의 다양성이 필요하기 때문이다.

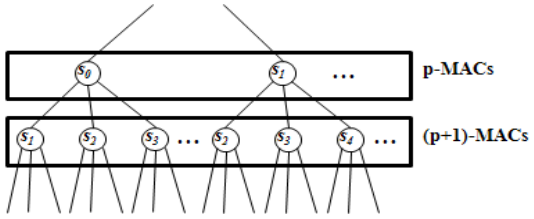
예를 들어, 두 p-MAC C_1의 O = {g_0, g_2, g_3, g_5}이고, C_2의 O = {g_1, g_2, g_3, g_4, g_6}이라고 할 때, C_1과 C_2간의 유사도는 0.5이다. 이 경우 rt가 0.4라면 C_1과 C_2는 유사하다고 할 수 있고, 0.6이라면 유사하다고 할 수 없다. 그러므로 rt가 높을수록 두 Macro-Cluster들이 유사하지 않을 확률이 높아지고, 따라서 많은 결과가 생성된다.

3. 알고리즘

정의 2에 따르면 Macro-Cluster는 샘플의 집합 하에서 비슷한 움직임을 보이는 유전자 집합을 말한다. Macro-Cluster를 마이닝하기 위해서 가장 먼저 하는 수행하는 프로세스는 1-MAC의 집합을 구하는 것이다. 정의3에서 그 과정이 설명되어 있다. 1-MAC 집합에서, 1) 유전자의 개수가 mg보다 작은 1-MAC과 2) 유전자 집합이 다른 유전자 집합에 포함되는 1-MAC은 제거된다. 그러므로 정의3에서의 1-MAC의 예인 ({g_3, g_6}, {s_0}), ({g_3, g_6, g_7, g_9}, {s_0}), ({g_7, g_9}, {s_0}), ({g_2}, {s_0}), ..., ({g_0, g_2, g_7}, {s_1}), ...에서, ({g_2}, {s_0})는 mg = 2 일 때 제거되고, ({g_3, g_6}, {s_0}), ({g_7, g_9}, {s_0})은 {g_3, g_6} 및 {g_7, g_9}가 {g_3, g_6, g_7, g_9}에 포함되므로 제거된다.

모든 1-MAC에 대해서 검사 가능한 샘플을 검사함으로써 2-MAC을 얻어낼 수 있다. 2-MAC에서 3-MAC을 얻어내고, 3-MAC에서 4-MAC을 얻어내는 과정도 이와 같다. 일반적으로 p ≥ 1 일 때, 각 p-MAC $C = (O, T)$에 대해서, 검사 가능한 샘플을 검사함으로써 (p+1)-MAC을 얻어낼 수 있다.

이러한 프로세스는 하나의 p-MAC이 레벨 p의 한 노드(node)인 넓이 우선 트리(breadth first tree)를 구성해 나가는 것과 같다고 볼 수 있으며, (그림 3)에 이 프로세스가 설명되어 있다. (그림 3)의 각 노드는 p-MAC 혹은 (p+1)-MAC을 뜻하며, 노드의 이름은 p-MAC 혹은 (p+1)-MAC을 얻기 위해 조사해야 하는 샘플을 뜻한다. 이어지는 장에서 이상의 샘플 검사 알고리즘을 보다 자세히 서술한 후, 효율적인 넓이 우선 트리를 구성하기 위한 큐 알고리즘과 중복 검사 알고리즘을 자세히 설명하고, 마지막으로 전체적인 알고리즘에 대한 시간 복잡도를 계산한다.



(그림 3) p 〉 1 일 때, p-MAC에서 (p+1)-MAC을 도출

3.1 검사 알고리즘

이 장에서 설명할 알고리즘은 p ≥ 1 일 때 임의의 p-MAC $C = (O, T)$에 대해서 검사 가능한 샘플을 검사함으로써 (p+1)-MAC C'를 생성해내는 알고리즘이다. 샘플 s_{last}가 T의 마지막 원소일 때, 검사 가능한 샘플은 last < i를 만족하는 s_i이다. 예를 들어 3-MAC의 T = {s_0, s_2, s_3} 일 때, 검사 가능한 샘플은 s_4, s_5, ..., $s_{(n-1)}$ 이 된다.

만약 위의 예에서 s_1을 검사한다면, 4-MAC $C_l = (O_l, T_l)$의 T_l = {s_0, s_2, s_3, s_1}이 된다. 여기서 또 다른 3-MAC이 {s_0, s_1, s_2}를 샘플 집합으로 가질 때, 샘플 s_3 를 조사하는 경우를 생각해 보면, 생성되는 4-MAC $C_2 = (O_2, T_2)$ 의 T_2 = {s_0, s_1, s_2, s_3} 이다. 이때, $O_l = O_2$ 라면, $C_l = C_2$ 라고 할 수 있으므로 C_2 를 생성하는 작업은 중복 검사가 된다.

Macro-Clustering은 검사하는 샘플의 순서와 상관없이 동일한 샘플 집합에 대해서 동일한 유전자 집합을 생성해낼 수 있다. 즉, 위의 예에서 $O_1 = O_2$ 가 된다. 이것을 설명하기 위해 예를 들면, 샘플 s_0에서 같은 범위의 유전자들을 고른 후, 이 유전자들이 샘플 s_1에서 같은 범위에 있는지 검사하는 작업과 샘플 s_1에서 같은 범위의 유전자들을 고른 후, 이 유전자들이 샘플 s_0에서 같은 범위에 있는지 검사하는 작업은 결국 동일한 유전자 집합을 생성한다. 이러한 특성 때문에 Macro -Clustering은 결정적(deterministic)이라고 할 수 있으며, 검사 가능한 샘플의 인덱스는 샘플 집합의 마지막 샘플의 인덱스보다 커야 한다는 조건은 타당하다고 할 수 있다.

p-MAC $C = (O, T)$에서 생성 가능한 (p+1)-MAC $C' = (O', T')$이라고 하고, 이 때 검사 가능한 샘플 si을 샘플 집합으로 하는 1-MAC을 $C'' = (O'', \{s_i\})$이라고 하자. 그렇다면 (p+1)-MAC $C' = (O', T')$의 $O' = O \cap O''$이고, $T' = T \cup \{s_i\}$로 계산될 수 있다. 즉, (p+1)-MAC의 생성은 p-MAC과 1-MAC을 합치는 과정과 같다. 또한 도출된 (p+1)-MAC은 p-MAC에 비해서 샘플 집합은 더 커지고 유전자 집합은 대체적으로 더 작아진다.

도출된 (p+1)-MAC이 더 이상 없거나, 도출된 모든 (p+1)-MAC $C' = (O', T')$의 $|O'| < mg$ 라면 전체 프로세스는 종료된다.

3.2 큐 알고리즘

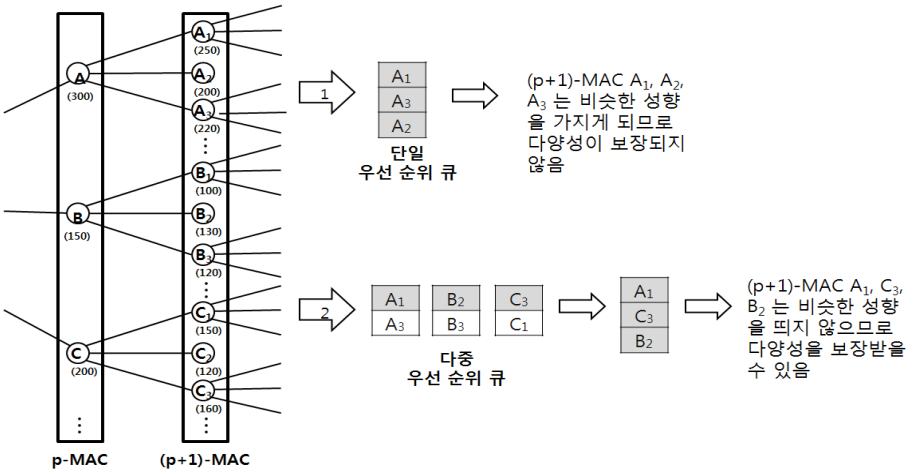
(그림 3)의 넓이 우선 트리에서, p가 커질수록 노드의 수는 기하급수적으로 증가한다. 모든 p-MAC의 개수가 r이라고 할 때, 각 p-MAC에 대해서 최대한 n개의 샘플이 조사될 수 있으며, 각 조사 프로세스는 해당 샘플에서의 binning된 유전자 집합의 개수만큼의 (p+1)-MAC을 생성할 수 있다. 해당 샘플에서의 binning된 유전자 집합 의 개수는 $O(m)$이므로 최대한 $O(mnr)$개의 (p+1)-MAC가 있다고 할 수 있다. 마이크로어레이가 조금만 커도, 메모리의 한계로 인해 서 이 많은 p-MAC을 저장할 수 없으며, 저장할 수 있다 하더라도 이 모든 p-MAC을 검사하는 것은 너무 많은 시간이 소요되는 작업이다. 그러나 다음과 같은 사실에 의해 모든 p-MAC을 검사할 필요가 없음을 알 수 있다.

1) 많은 p-MAC이 중복되어 있거나, 조금씩 밖에 다르지 않으므로, 유전자 집합이 보다 크고, 유일성이 있는 p-MAC에 관심이 더 있다.
2) $p' > p$인 p'에 대해서 p'-MAC으로 성장할 수 있는 가능성이 더 큰 p-MAC에 관심이 더 있다.

조건 1)을 만족하기 위해서, 넓이 우선 트리의 각 레벨마다 유전자 집합의 크기 $|O|$를 우선 순위 측정 함수로 하는 우선 순위 큐를 유지함으로써 p-MAC을 저장한다. 이 우선순위 큐의 p-MAC은 최종 결과로 사용된다. (그림 4)는 하나의 큐를 이용해서 p-MAC을 저장하는 전략과 다수의 큐를 이용하는 전략을 비교하고 있다. (그림 4)의 넓이 우선 트리의 각 노드는 Macro-Cluster를 의미하며, 노드의 괄호 안의 숫자는 유전자 집합의 크기를 나타낸다. 하나의 큐를 이용하는 전략은 p-MAC B와 C로부터 생성되는 (p+1)-MAC를 버리고 있으며, 이것은 이 전략이 다양성을 보장하지 않음을 보여준다. 그러므로 본 논문에서는 다수의 큐를 이용함으로써 다양성을 보장하고 있다.

조건 2)를 만족하기 위해서 위에서 설명했던 우선순위 큐를 한 세트 더 사용한다. 큐 저장 전략은 같지만, 우선순위 측정 함수는 $|O|$ x $(n - last)$ 로 하며, 여기서 $last$ 는 T의 마지막 샘플의 인덱스를 말한다. 예를 들어 3-MAC의 $T = \{s_0, s_2, s_3\}$ 일 때, $last$는 3이다. 이 우선 순위 측정 함수의 장점은, 1) 일반적으로 보다 큰 유전자 집합을 가진 p-MAC이 보다 큰 유전자 집합을 가진 (p+1)-MAC으로 성장할 수 있는 확률이 높으며, 2) $last$가 커질수록, p-MAC이 더 큰 MAC으로 성장할 확률이 적어진다는 것이다.

예를 들어 전체 샘플 집합 $S = \{s_0, s_1, s_2, s_3, s_4, s_5\}$일 때, 두 개의 3-MAC인 SB_1와 SB_2를 가정해보자. 이고, SB_1의 $T = \{s_0, s_1, s_2\}$이고, SB_2의 T = $\{s_0, s_1, s_3\}$이다. 이 때 SB_1이 SB_2보다 검사할 수 있는 샘플을 많이 가진다(SB_1의 경우 s_3, s_4, s_5이고, SB_2는 s_4, s_5이다). 이것은 SB_1이 T =



(그림 4) 단일 우선순위 큐와 다중 우선순위 큐의 비교

{s_0, s_1, s_2, s_3}, T = {s_0, s_1, s_2, s_4}, T = {s_0, s_1, s_2, s_5}인 3개의 4-MAC으로 성장할 수 있음에 비해서, SB_2는 T = {s_0, s_1, s_2, s_3, s_4}, T = {s_0, s_1, s_3, s_5}인 2개의 4-MAC으로 밖에는 성장할 수 없음을 뜻한다. 또한 SB_1는 T = {s_0, s_1, s_2, s_3, s_4, s_5}인 6-MAC으로 성장할 수 있음에 비해서 SB_2는 6-MAC으로 성장할 수 없다. 그러므로 *last*가 커질수록, p-MAC이 성장할 수 있는 확률은 줄어듦을 예상할 수 있고, 위의 우선순위 측정 함수를 사용할 경우, 우선순위 큐는 더욱 큰 MAC을 담을 수 있음을 알 수 있다.

위에서 설명한 두 세트의 우선순위 큐를 사용할 경우, 모든 p-MAC을 조사한 후, 두 세트의 (p+1)-MAC을 얻을 수 있다. 한 세트는 결과 Macro-Cluster가 되며, 다른 세트는 다음 (p+1)-MAC를 생성하기 위한 후보가 된다.

각 우선 순위 큐의 크기를 *qsize*라고 하고, 우선순위 큐의 개수를 *qnum*이라 하자. 이 두 변수는 사용자 입력값이 될 수 있다. *qsize*는 결과의 다양성에 영향을 미친다. 본 논문에서는 *qsize*를 내부적으로 적당히 큰 값인 100으로 고정시켰는데, 그 이유는 *qsize*가 결과의 다양성의 정도와 비례하는 것은 아니기 때문이다. 즉, *qsize*는 아주 작지만 않으면 다양성을 해치지 않기 때문이다. k = *qnum* x *qsize* 라고 할 때, k 는 중복된 p-MAC를 제거하기 전의 p-MAC의 총 개수가 된다. *qnum*이 클수록 k 또한 커지며, 일반적으로 k가 클수록 가지치기(pruning)을 방지하므로 최적의 해를 찾을 확률을 높일 수 있다. 그러나 <표 5>의 실험 결과에서 *qnum*이 100보다 크다면, *qnum*이 바이클러스터의 품질에 영향을 미치지 않음을 확인 할 수 있다.

3.3 중복 Macro-Cluster의 정리

유사하지 않은 (p+1)-MAC을 얻기 위해서 두 우선 순위 큐 각각의 (p+1)-MAC에 대한 중복 검사를 수행한다. 중복 검사의 첫 번째 단계는 모든 (p+1)-MAC에 대해서 정의 4에 서술된 유사도 검사를 통해 유사도 점수를 부여함으로써 유사한 (p+1)-MAC끼리 클러스터링하는 것이다. 유사한 (p+1)-MAC의 클러스터에서 유전자 집합의 크기인 |T|가 가장 큰 (p+1)-MAC를 골라낸다. 선택되지 못한 (p+1)-MAC는 일정 정도 이상 중복된 것을 의미하므로 버릴 수 있다. 이상 설명된 전반적인 알고리즘이 (그림 5)에 나와 있다.

3.4 알고리즘의 시간 복잡도

마이크로어레이 데이터의 한 열에 대한 정규화 작업은 O(m)의 시간이 소요되며, 열의 개수는 n개 이므로, 정규화 작업은 전체적으로 O(nm)의 시간이 소요된다.

각 열에 대해서, 생성될 수 있는 범위의 개수는 유전자의 개수를 넘지 않으므로 O(m)개 이라고 할 수 있다. 그러나 실제로 범위의 개수는 훨씬 작다. 각 유전자가 어떤 범위에 속하는지 검사하고 binning하는 작업은 범위의 개수와 유전자 개수의 곱인 O(m^2)의 시간이 소요된다. 따라서 binning 프로세스의 전체적인 시간 복잡도는 O(nm^2)이다.

p-MAC에서 (p+1)-MAC를 생성하는 작업은, 검사 가능한 모든 샘플에 대해서 해당 샘플을 가지는 1-MAC과 p-MAC을 합치는 작업과 같으며, 이것은 두 Macro -Cluster의 유전자 집합의 교집합을 구하는 시간과 같은 O(m)이 소요된다. 검사 가능한 샘플의 수는 O(n)이며, 각 샘플에서의 1-MAC의 개수는 범위의 개수와 같은 O(m)이므로 전체적으로 O(nm^2)의 시간이 소요된다.

p-MAC의 개수는 최대로 큐의 크기를 넘을 수 없다. 이 때, 큐의 크기는 100 x *qnum*이고, 앞서 설명했듯이 p-MAC에서 (p+1)-MAC을 생성하는 작업은 O(nm^2)의 시간이 소요되므로, 모든 p-MAC에서 모든 (p+1)-MAC을 생성하는 작업은 O(*qnum* x nm^2)의 시간이 소요된다.

p > 1인 모든 p-MAC을 생성하는 작업은 최대한 전체 샘플의 개수인 O(n)만큼 반복된다. 따라서 이 작업의 전체 시간 복잡도는 O(*qnum* x n^2m^2)이고, 전체적인 Macro-Clustering의 시간 복잡도는 O(nm) + O(nm^2) + O(*qnum* x n^2m^2) = O(*qnum* x n^2m^2)으로 계산된다. 즉, Macro-Clustering은 지수적인 시간 복잡도를 가지지 않음을 알 수 있다.

입력값 : D = (G, S)의 마이크로어레이, *mg*, *ms*, *a*, *d*, *rt*, *qnum*
출력값 : p-MAC 집합 (p ≥ *ms*)
1: 마이크로어레이 데이터의 각 열을 정규화
2: 정규화된 각 열에 대해서 3~5를 반복
3: 열의 최소값이 속하는 범위가 R_{min}이고 열의 최대값이 속하는 범위가 R_{max}일 때, min<i<max인 정수 i에 대해서 R_i를 모두 계산(정의 1 참조)
4: 열의 각 값을 해당되는 범위 R_i에 binning
5: 해당 열의 샘플이 s_i라고 할 때, 1-MAC C = (각 범위에 binning된 유전자 집합, {s_i})를 임시 저장소에 저장
6: p ← 1;
7: |T| ≥ *mg* 인 p-MAC이 더 이상 생성되지 않을 때까지 8~15를 반복
8: 임시 저장소에 저장된 모든 p-MAC C = (O, T)에 대해서 9~10을 반복
9: 샘플 s_i를 가지는 1-MAC C'' = (O'', {s_i})라고 할 때, O' = O ∩ O'' 이고, T' = T ∪ {s_i} 인 (p+1)-MAC C' = (O', T')를 생성
10: C'의 |T'| ≥ *mg* 라면 C'를 결과 우선순위 큐 및 후보 우선순위 큐에 저장함
11: *rt*에 의해서 두 큐에 저장된 중복된 (p+1)-MAC를 제거
12: (p+1)≥ms 라면 결과 우선순위 큐에 저장된 모든 (p+1)-MAC을 출력
13: 후보 우선순위 큐에 저장된 모든 (p+1)-MAC을 임시 저장소에 저장
14: 두 큐를 비움
15: p ← p+1;

(그림 5) Macro-Clustering 알고리즘

4. 실험 및 결과

본 실험에서는 실제 마이크로어레이 데이터에 대해서 파라미터를 변화시켜 나가면서 도출된 Macro-Cluster의 기능적 상관관계의 정도를 측정함으로써 최적의 파라미터를 정하는 방법을 서술한다. 또한 실제 데이터에 대해서 다른 알고리즘이 도출하는 바이클러스터와 비교해서 Macro-Cluster가 어느 정도의 생물학적인 의미를 가지고 있는가를 알아본다.

4.1 실험 환경

본 실험에서 사용하는 실제 데이터는 2가지이다. 하나는 Tavazoie[13]의 yeast의 2884개의 유전자를 17개의 샘플에 대해 실험한 마이크로어레이이며, 다른 하나는 Gasch[14]의 yeast의 2994개의 유전자를 173개의 샘플에 대해 실험한 마이크로어레이이다. yeast는 비교적 유전자의 기능이 많이 밝혀져 있기 때문에, yeast의 유전자의 발현값을 나타내는 마이크로어레이에 대해서 바이클러스터링을 수행한 결과 높은 신뢰도로 같은 기능을 하는 유전자 집합을 클러스터링할 수 있다면, 아직 기능이 밝혀지지 않은 인간 등의 고등 생물의 유전자에 대해서도 좋은 결과를 생성해 낼 수 있다고 기대할 수 있다.

본 Macro-Clustering 알고리즘은 Standard Template Library를 사용한 C++로 구현되었으며, Windows XP 운영체제에서, 1GB의 메모리와 AMD Athlon 64 X2 Dual 2.81GHz의 CPU를 가진 컴퓨터를 사용해 실험했다. 이어지는 모든 실험의 경우에 별다른 언급이 없을 경우 ms = 4, mg = 10, rt = 0.5, $qnum$ = 100, d = 2를 사용하고, 2994 x 173 yeast 데이터의 경우 a = 0.15를, 2884 x 17 yeast 데이터의 경우 a = 0.10을 사용했다.

4.2 파라미터 결정

본 장에서는 Macro-Clustering의 파라미터인 a, d 및 $qnum$의 결정을 위해서 각 파라미터를 변화시켜가면서 결과 Macro-Cluster의 기능적 상관 관계 정도를 측정한다. 또한 파라미터 rt의 결정을 위해서 rt를 변화시켜 나가면서 생성되는 Macro-Cluster의 개수를 측정한다.

Macro-Cluster의 기능적 상관관계의 정도는 유전자 온톨로지(Gene Ontology, 이하 GO) 데이터베이스의 검색을 통해서 이루어진다. GO 데이터베이스는 임의의 생물 종에서 나타나는 세포 단위의 기능을 계층도 형식으로 나타낸 데이터베이스를 뜻한다. 각 기능에 참여하는 유전자의 목록 또한 기능과 함께 GO 데이터베이스에 기록된다. 본 실험에서는 상기 설명한 두 가지 실제 yeast 마이크로어레이에 대한 Macro -Cluster를 도출해서, Macro-Cluster에 참여하는 유전자가 임의의 기능에 참여하는 양상을 GO 데이터베이스의 검색을 통해서 살펴봄으로써 그 기능적 상관관계 정도를 측정하고자 한다.

이 작업은 FuncAssociate[15]을 통해서 수행한다. FuncAssociation은 유전자 집합을 입력값으로 해서 가장 최근의 GO 데이터베이스 검색을 자동으로 수행하고, 유의한 기능들 및 각 기능 하에서의 유전자의 기능적 상관관계의 정도를 통계화해서 출력해주는 자동화된 툴이다. FuncAssociate의 결과로 해당 바이클러스터의 유전자 집합에 대한 p-value가 계산된다. 이 p-value는 유전자 집합이 같은 기능을 하지 않는다는 귀무가설을 기각하기 위한 유의 수준의 최소값을 말한다. 그러므로 p-value가 작을수록 유전자 집합이 같은 기능을 한다고 말할 수 있다.

본 실험에서는 a, d 및 $qnum$을 변화시켜 나가면서 Macro-Clustering을 수행했다. 그 후 결과 Macro -Cluster를 무작위로 추출하여 p-value를 검사하고, 샘플의 수가 10개인 Macro-Cluster들의 유전자 개수의 평균을 측정했다. 유전자 개수의 평균을 측정한 이유는 파라미터에 따라서 달라지는 Macro-Cluster의 크기를 확인하기 위해서이다.

<표 3> (a), (b), (c) 및 (d)는 a 및 d 값이 어느 정도 이상이 되면 극히 낮은 p-value를 보여주는 Macro-Cluster를 생성할 수 있음을 보여준다. 낮은 p-value를 가지는 Macro-Cluster를 생성할 수 있는 a 및 d 값은 매우 넓은 범위를 보여주고 있으므로, 파라미터의 선택의 폭이 넓다는 장점이 된다. 더군다나 겉으로 보기에 아주 상이한 분포를 가지는 두 yeast 데이터 모두에 비슷한 a 및 d 값을 사용할 수 있는 것을 볼 수 있는데, 그것은 이 두 데이터가 정규화되었기 때문이라고 판단되며, 실제 많은 수의 상이한 마이크로어레이에 대해서 파라미터의 선택을 쉽게 할 수 있는 장점이 된다.

이 실험 결과로써, 임의의 마이크로어레이에 대해서 a 와 d는 각각 최소한 0.05 및 1.7이 넘는다면 유의한 Macro-Cluster를 생성할 수 있다고 기대할 수 있다. 그 밖에 각 파라미터를 정할 때 기준이 될 수 있는 것은 각 값 하에서의 유전자의 개수이다. 임의의 마이크로어레이 에 대해 많은 유전자가 참여하는 기능, 즉 상위 기능을 밝혀야 하는 경우라면 될 수 있는 한 a 와 d를 크게 해주는 것이 유리하고, 하위 기능을 밝혀야 하는 경우라 면 될 수 있는 한 a 와 d를 작게 해주는 것이 유리하다고 판단된다.

<표 3> (e)는 qnum이 50이 넘을 경우 Macro-Cluster의 p-value가 그리 차이가 나고 있지 않으며, 특별히 더 큰 유전자 집합을 찾아내고 있지 못함을 보여준다. 더군다나 앞서 $qnum$의 증가에 따라서 전체 실행 시간이 선형적으로 증가함을 보았을 때, $qnum$을 100이상으로 하는 것은 필요하지 않다고 판단된다.

마지막으로 rt를 변화시켜 나가면서 Macro- Clustering을 수행해서 결과 바이클러스터의 개수를 확인해보았다. (그림 6)에서 rt가 증가할수록 결과 Macro-Cluster의 개수가 기하급수적으로 늘어나는 것을 볼 수 있다. rt가 증가할수록 많이 중복된 바이클러스터를 생성하며, 이 바이클러스터들은 서로 간에 조금씩만 다를 뿐이므로 그 양 또한 많아진다. 이러한 바이클러스터 모두를 검사하는 것은 큰 비용이 필요하다. (그림 6)은, 중복 검사를 하지 않을 경우, 조금씩만 다른 매우 많은 Macro-Cluster가 생성될 수 있다는 것을 보여준다.

〈표 3〉 *a*, *d* 및 *qnum*을 변경하면서 측정한 p-value와 유전자의 개수

a	0.01	0.03	0.05	0.07	0.085	0.1	0.115	0.13
p-value	7.7E-41	3.5E-72	3.6E-75	7.6E-67	1.0E-90	8.6E-99	2.0E-81	2.6E-91
#유전자	17	72	124	267	279	314	307	398
A	0.15	0.17	0.2	0.25	0.3	0.4	0.5	0.7
p-value	4.9E-93	2.8E-66	4.2E-77	2.6E-89	8.3E-95	1.4E-61	2.3E-65	1.5E-74
#유전자	524	447	433	869	415	694	844	1125

(a) 2884 x 17 yeast 데이터에 대해 d = 2로 고정하고 a를 변경

d	1.5	1.7	1.85	2	2.15	2.3	2.5
p-value	6.5E-43	2.8E-88	4.3E-88	8.6E-99	3.0E-84	3.1E-92	5.1E-90
#유전자	48	169	201	314	468	512	568
d	3	4	5	7			
p-value	1.1E-76	3.6E-54	2.9E-50	2.9E-50			
#유전자	712	924	990	990			

(b) 2884 x 17 yeast 데이터에 대해 a = 0.1로 고정하고 d를 변경

a	0.01	0.05	0.10	0.15	0.20	0.40
p-value	3.8E-08	1.3E-41	1.1E-111	4.3E-130	1.3E-101	7.4E-53
#유전자	26	26	36	55	91	133

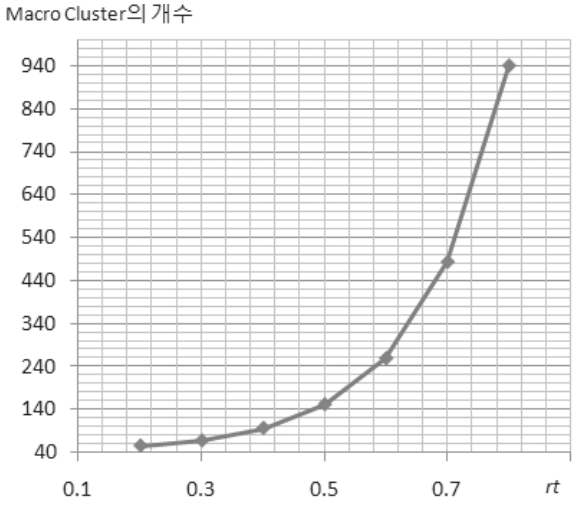
(c) 2994 x 173 yeast 데이터에 대해 d = 2로 고정하고 a를 변경

d	1.5	1.8	2.0	2.2	2.5	3.0
p-value	5.3E-12	2.8E-68	4.3E-130	3.9E-57	3.1E-77	4.6E-82
#유전자	19	28	55	65	209	240

(d) 2994 x 173 yeast 데이터에 대해 a = 0.15로 고정하고 d를 변경

qnum	10	50	100	200	300	500	1000
p-value	5.1E-06	1.1E-92	8.6E-99	4.7E-95	4.7E-95	2.4E-93	2.4E-93
#유전자	307	310	314	314	314	315	315

(e) 2884 x 17 yeast 데이터에 대해 a = 0.1, d = 2로 고정하고 *qnum*을 변경



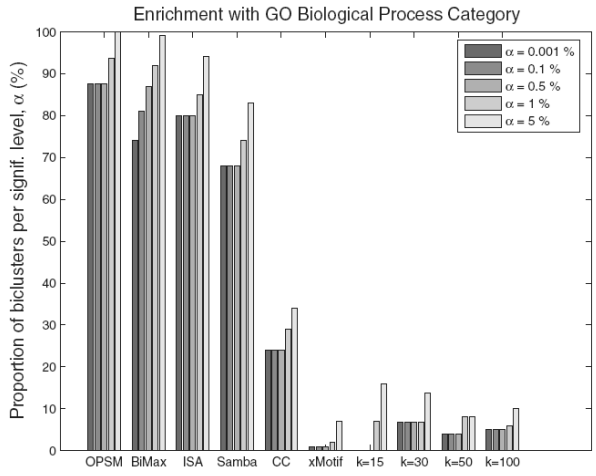
(그림 6) 2884 x 17 yeast 데이터에 대해 *rt*의 변화에 따른 Macro-Cluster의 개수

4.3 Macro-Cluster의 생물학적 의미 및 기능적 상관 관계 정도의 측정과 비교

(그림 7)은 Prelic[9]에서 OPSM[2], BiMax[9], ISA[16], Samba[18], CC[3], xMotif[19]와 기존의 계층적 클러스터 링 기법을 이용해서 도출된 유전자 집합의 유의 수준을 측정한 것이다. (그림 7)에서의 비교 알고리즘은 모두 2994 x 173 yeast 마이크로어레이에서 실행되었으며, α는 p-value와 같다. (그림 7)에서 OPSM의 결과 바이클러스터 중 90%가 유의 확률 0.001%보다 작은 것을 알 수 있다. 이로서 OPSM이 기타 알고리즘보다 더욱 기능적 상관관계의 정도가 높은 결과를 도출해 냄을 알 수 있다.

Macro-Clustering이 어느 정도의 확률로 0.001%보다 작은 Macro-Cluster를 도출하는지 알아보기 위해서 (그림 7)에서와 같은, 2994 x 173 yeast 마이크로어레이에 대해서 Macro-Clustering을 수행했다. 그 결과 도출된 모든 Macro-Cluster가 0.001%보다 작은 p-value를 가짐을 확인할 수 있었다. 따라서 Macro-Clustering은 OPSM보다 좋은 결과를 보여준다고 말할 수 있다. 본 실험에서는 더욱 정확한 결과를 위해서 2994 x 173 yeast 마이크로어레이 뿐만 아니라 2884 x 17 yeast 마이크로어레이에 대해서도 OPSM과 Macro-Clustering을 모두 실행시킨 후 결과 바이클러스터에 대해서 FuncAssociation을 통해 p-value를 살펴보았다.

OPSM을 실행하기 위해서 BicAt v2.22 [17]을 이용했으며, BicAt에서 구현한 OPSM 알고리즘의 iteration 파라미터는 기본 제공값인 10을 사용했다. 두 yeast 마이크로어레이에 대해서 OPSM은 각각 14개의 바이클러스터를 생성했으며, 각각의 경우에서 낮은 p-value를 가지는 바이클러스터 1개씩을 선택한 후, 유전자의 개수가 선택된 바이클러스터와 최대한 비슷한 Macro-Cluster 1개씩을 다시 선택해서 비



(그림 7) 바이클러스터링 알고리즘 및 계층적 클러스터링 알고리즘의 결과 유전자 집합의 유의 확률 별 분포

교하는 방식으로 실험을 진행했다. 그 결과는 <표 4>와 같다. <표 4>의 (a)와 (b)는 2884 x 17 yeast 데이터에 대해서, (c)와 (d)는 2994 x 173 yeast 데이터에 대해서 유전자의 개수가 비 슷한 Macro-Cluster와 OPSM의 바이클러스터에 대한 GO 검증 결과이다. 또한 <표 4>의 (e)는 (a) ~ (d)를 해석하기 보기표이며, (f)은 각 (a) ~ (d)에서 나타난 각 GO에 대한 설명을 나타낸다.

<표 4>의 실험에서 알 수 있듯이 전체적으로 Macro-Clustering은 OPSM과 비슷한 GO에서 훨씬 낮은 p-value를 보여주고 있다. Macro-Clustering은 2994 x 173 yeast 데이터의 경우에는 전체적으로 약간씩 낮은 p-value를 보여주고 있으며, 특히 2884 x 17 yeast 데이터의 경우에는 훨씬 다양한 GO에 대해서 크게 낮은 p-value를 보여주는 바이클러스터를 찾고 있음을 확인할 수 있다. (그림 8)의 (a) 및 (b)는

〈표 4〉 FuncAssociate를 이용한 GO 검증 결과

Rank	N	X	P	GO 이름
1	71	168	2.2e-92	GO:0005830
2	71	186	2.0e-88	GO:0044445
3	76	276	3.6e-83	GO:0005840
4	69	236	6.4e-76	GO:0033279
5	68	230	5.5e-75	GO:0003735

(a) 115 x 15 Macro-Cluster

Rank	N	X	P	GO 이름
1	3	7	4.5e-05	GO:0000407

(b) OPSM의 82 x 8 바이클러스터

Rank	N	X	P	GO 이름
1	44	168	1.7e-71	GO:0005830
2	44	186	2.9e-69	GO:0044445
3	44	230	1.0e-64	GO:0003735
4	44	236	3.6e-64	GO:0033279
5	44	276	6.8e-61	GO:0005840

(c) 45 x 19 Macro-Cluster

Rank	N	X	P	GO 이름
1	42	168	4.3e-68	GO:0005830
2	42	186	5.6e-66	GO:0044445
3	42	230	1.2e-61	GO:0003735
4	42	236	3.8e-61	GO:0033279
5	42	276	4.9e-58	GO:0005840

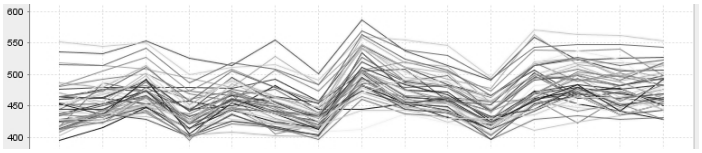
(d) OPSM의 43 x 11 바이클러스터

Rank	바이클러스터의 유전자 집합의 P-value에 의한 GO의 순위
N	바이클러스터의 유전자 집합 중 GO에 속하는 유전자의 개수
X	GO를 이루는 유전자의 개수

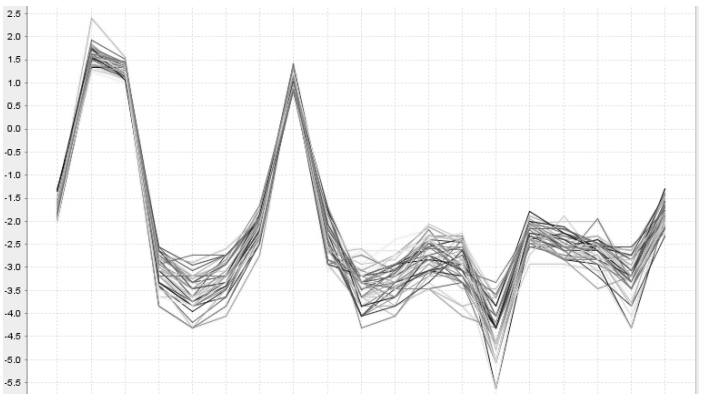
(e) Table 보기표

GO 이름	GO에 대한 설명
GO:0000407	pre-autophagosomal structure
GO:0003735	structural constituent of ribosome
GO:0005830	cytosolic ribosome (sensu Eukaryota)
GO:0005840	ribosome
GO:0033279	ribosomal subunit
GO:0044445	cytosolic part

(f) GO에 대한 보기표



(a) 2884 x 17 yeast 데이터의 결과 115 x 15 Macro-Cluster



(b) 2994 x 173 yeast 데이터의 결과 45 x 19 Macro-Cluster

(그림 8) Macro-Cluster의 유전자 그래프

각각 <표 4>의 (a) 및 (c)의 Macro-Cluster를 유전자 그래프로 표시한 것이다.

5. 결 론

바이클러스터는 같은 기능을 하는 유전자의 집합과, 그 기능이 발현되는 샘플의 집합이다. 만약 바이클러스터를 구성하는 유전자 중에서 일부 유전자의 기능이 A라고 알려져 있다면, 나머지 기능이 알려지지 않은 유전자의 기능은 A라고 유추할 수 있다. 이것은 실제적인 생물학적 실험의 범위를 줄여줌으로써 그 비용을 절감하게 하는 효과를 지닌다. 또한, 샘플의 집합에서도 유사한 기능의 발현을 관찰할 수 있다는 점에서 비슷하게 이용될 수 있다. 예를 들어 암환자와 정상인이 샘플로 섞인 마이크로어레이 데이터에서 암환자 샘플만으로 구성된 바이클러스터의 특성을 알 수 있다면, 추후에 암환자인지 정상인인지 알 수 없는 환자의 암 진단을 쉽게 할 수 있다.

마이크로어레이 데이터는 실험적 오차가 존재할 수 있는 데이터를 수치화한 것이기 때문에, 비교적 노이즈가 높다는 단점이 있다. 이것은 마이크로어레이 데이터의 노이즈를 잘 다룰 수 없다면, 의미 있는 데이터를 마이닝할 수 없음을 말한다. 또한 마이크로어레이 데이터는 그 유전자 및 샘플의 개수가 많은 경우가 대부분이며, 이 데이터를 잘라낸 일부분만으로는 의미 있는 데이터를 마이닝할 수 없는 경우도 많다. 따라서 샘플 및 유전자의 개수에 대해서 지수적인 시간 복잡도를 가지는 알고리즘은 바람직하지 않다. 그리고 마이크로어레이 실험은 상대적으로 고가이므로, 한 마이크로어레이 데이터에서 최대한 많고 다양한 바이클러스터를 찾는 것이 유리하다.

본 연구에서 제안하는 Macro-Clustering 알고리즘은 거시적으로 비슷한 패턴을 가지는 유전자의 집합을 클러스터링 함으로써 실험적인 영향에 기반한 노이즈에 대해서 견고한 특징을 보인다. 또한 효율적인 큐 알고리즘을 통해서 효과적으로 넓이 우선 트리를 구성해 나감으로 인해 실용적인 시간 복잡도를 가진다. 큐 알고리즘과 중복 제거 알고리즘은 시간 효율성뿐만 아니라 결과의 다양성 또한 보장해준다. 마지막으로, Macro-Cluster가 높은 생물학적, 기능적 상관관계를 보임을 증명했다.

향후 연구로, 바이클러스터 간의 상관성을 연구함으로써, 여러 다양한 조건 하에서 생물학적 pathway를 찾는 기법을 개발할 예정이며, 암환자의 마이크로어레이 데이터를 통해 암의 진행 정도 등을 효과적으로 진단할 수 있는 바이클러스터의 활용 등을 연구할 계획이다.

참 고 문 헌

[1] S. C. Madeira and A. L. Oliveira, “Biclustering Algorithms for Biological Data Analysis: A Survey,” IEEE/ACM Trans. Computational Biology and Bioinformatics, Vol.1, No.1, pp. 24-45, 2004.

[2] A. Ben-Dor, B. Chor, R. Karp, and Z. Yakhini, “Discovering local structure in gene expression data: The order-preserving submatrix problem,” in Proc. 6th Int’l Conf. Computational Biology, pp.49-57, 2002.

[3] Y. Cheng and G.M. Church, “Biclustering of Expression Data,” in Proc. 8th Int’l Conf. Intelligent Systems for Molecular Biology, pp.93-103, 2000.

[4] W. Liu and L. Chen, “A Fast Algorithm for Gene Expressing Data Biclustering,” International Journal of Intelligent Information Technology Application, Vol.1, issue1, pp.30-36, 2008.

[5] H. Wang, W. Wang, J. Yang and P. S. Yu, “Clustering by Pattern Similarity in Large Data Sets,” in Proc. ACM SIGMOD Int’l. Conf. Management of Data, pp.394-405, 2002.

[6] L. Zhao and M. J. Zaki, “triCluster: An Effective Algorithm for Mining Coherent Clusters in 3D Microarray Data,” in Proc. ACM SIGMOD Int’l. Conf. on Management of data, pp.694-705, 2005.

[7] X. Xu, Y. Lu, A. K. H. Tung and W. Wang, “Mining Shifting-and-Scaling Co-Regulation Pattern on Gene Expression Profiles,” in Proc. 22nd IEEE Int’l. Conf. on Data Engineering, pp.89-99, 2006.

[8] X. Liu and L. Wang, “Computing the maximum similarity bi-clusters of gene expression data,” Bioinformatics, Vol.18, No.1, pp.50-56, 2007.

[9] A. Prelic, S. Bleuler, P. Zimmermann, A. Wille, P. Bhlmann, W. Gruissem, L. Hennig, L. Thiele, and E. Zitzler, “A systematic comparison and evaluation of biclustering methods for gene expression data,” Bioinformatics, Vol.22, No.9, pp.1122-1129, 2006.

[10] J. Liu and W. Wang, “Op-cluster: Clustering by tendency in high dimensional space,” in Proc. IEEE Int’l. Conf. on Data Mining, pp.187-194, 2003.

[11] B. J. Gao, O. L. Griffith, M. Ester, and S. J. M. Jones, “Discovering significant OPSM subspace clusters in massive gene expression data,” in Proc. 12th ACM SIGKDD pp.922-928, 2006.

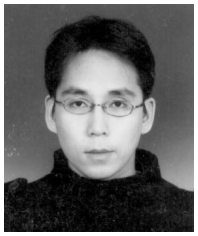
[12] Y. Zhao, G. Wang, Y. Yin and G. Yu, “Mining Positive and Negative Co-regulation Patterns from Microarray Data,” in Proc. 6th IEEE Symposium on BioInformatics and Bio-Engineering, pp.86-93, 2006.

[13] S. Tavazoie, J. D. Hughes, M. J. Campbell, R. J. Cho, and G. M. Church, “Systematic determination of genetic network architecture,” Nature Genetics, Vol.22, pp.281-285, 1999.

[14] A. P. Gasch, P. T. Spellman, C. M. Kao, O. Carmel-Harel, M. B. Eisen, G. Storz, D. Botstein and P. O. Brown, “Genomic expression programs in the response of yeast cells to environmental changes,” Molecular Biology of the Cell, Vol.11, pp.4241-57, 2000.

[15] G. F. Berriz, O. D. King, B. Bryant, C. Sander and F. P. Roth,

"Characterizing gene sets with FuncAssociate," Bioinformatics, Vol.19, No.18, pp.2502-2504, 2003.
[16] J. Ihmels, S. Bergmann and N. Barkai, "Defining transcription modules using large-scale gene expression data," Bioinformatics, Vol.20, No.13, pp.1993-2003, 2004.
[17] S. Barkow, S. Bleuler, A. Prelic, P. Zimmermann and E. Zitzler, "BicAT: a biclustering analysis toolbox," Bioinformatics, Vol.22, No.10, pp.1282-1283, 2006.
[18] A. Tanay, R. Sharan and R. Shamir, "Discovering statistically significant biclusters in gene expression data," Bioinformatics, Vol.18, No.1, pp.136-144, 2002.
[19] T. M. Murali and S. Kasif, "Extracting conserved gene expression motifs from gene expression data," Pac. Symp. Biocomput., 8, 77-88, 2003.
[20] J. Han, M. Kamber, Data Mining: Concepts and Techniques, second ed., Morgan Kaufman, San Francisco, CA, 2006.



안 재 균

e-mail : ajk@cs.yonsei.ac.kr
2006년 연세대학교 컴퓨터과학(학사)
2007년~현 재 연세대학교 컴퓨터과학과 석사과정
관심분야 : 데이터베이스 시스템, 데이터 마이닝, 바이오인포매틱스

윤 영 미

e-mail : ymyoon@gachon.ac.kr,
1981년 서울대학교(학사)
1981~1983년 오하이오 주립대학 수학과 (학사수료)
1987년 스탠포드대학교 컴퓨터과학과(석사)
1987~1993년 IntelliGenetics Inc., Mountainview, California, Software Engineer
2008년 연세대학교 컴퓨터과학과(박사)
1995년~현 재 가천의과학대학교 부교수
관심분야 : 데이터베이스 시스템, 데이터 마이닝, 바이오인포매틱스

박 상 현

e-mail : sanghyun@cs.yonsei.ac.kr
1989년 서울대학교 컴퓨터공학과(학사)
1991년 서울대학교 컴퓨터공학과(석사)
2001년 UCLA 컴퓨터과학과(공학박사)
2002년~2003년 포항공과대학교 컴퓨터공학과 조교수
2003~2006년 연세대학교 컴퓨터과학과 조교수
2006~현 재 연세대학교 컴퓨터과학과 부교수
관심분야 : 데이터베이스, 데이터 마이닝, 바이오인포매틱스, 적응적 저장장치 시스템