DNA 마이크로어레이 데이터의 계층적 클러스터링에 대한 리프오더링 알고리즘 개발[1)]

여상수[0] 이정원 김성권
중앙대학교 컴퓨터공학과
{ssyeo[0], biomania}@alg.cse.cau.ac.kr
skkim@cau.ac.kr

A Heuristic Leaf Ordering Algorithm for Hierarchical Clustering of DNA Microarray Data

Sang-Soo Yeo[0] Jung-Won Rhee Sung-Kwon Kim
Dept. of Computer Science & Engineering, Chung-Ang Univ.

요 약

DNA 마이크로어레이 실험으로 나온 데이터들을 클러스터링하는 것은 유전자의 기능과 유전자의 네트워크를 파악해 나가는데 도움을 주게 된다. 계층적 클러스터링(hierarchical clustering) 방법은 그러한 실험 분석에서 가장 보편적으로 사용되는 방법이다. 본 논문에서는 계층적 클러스터링을 통해서 나온 결과 트리에 대해서, 트리의 리프 노드들을 재배열함으로써, 인접한 리프 노드들간의 거리의 종합이 최소가 되도록 하는 문제인 리프오더링 방법을 다루었고, 새로운 리프오더링 알고리즘을 제안하였다. 그리고, 이를 포함한 여러 리프오더링 방법들에 대한 실험 및 생물학적인 분석을 하였다.

1. DNA 마이크로어레이 데이터의 클러스터링

DNA 마이크로어레이 실험 데이터는 일반적으로 $m \times n$행렬 D로 나타낼 수 있다. 여기서 행의 개수 m은 유전자의 개수로서 일반적으로 $10^3 \sim 10^4$의 범위를 갖는다. 반면에 열의 개수 n은 마이크로어레이의 실험의 횟수로써 대략 $10^1 \sim 10^2$의 범위를 갖는다[1]. DNA 마이크로어레이 데이터를 클러스터링하는 문제가 어려운 이유는 클러스터링해야 되는 유전자의 개수 m에 비해, 실험한 횟수 n이 상대적으로 매우 적은 수이기 때문이다. 이 문제에 대한 현재까지의 연구결과는 [1,2]의 논문에 잘 나와 있다.

본 논문에서는 다양한 클러스터링 방법 중에서 가장 일반적으로 생물학자들에 의해서 사용되고 있는 계층적 클러스터링(hierarchical clustering) 알고리즘을 연구 대상으로 하였다. 계층적 클러스터링의 결과물은 덴드로그램(dendrogram)으로서, 노드들에 대한 트리 구조와 노드들간의 유사도(또는 거리)를 함께 표현한 것이다. 본 논문에서는 계층적 클러스터링 알고리즘의 결과 트리의 리프 노드(유전자를 의미함)의 순서를 재배열하는 방법(리프오더링이라고 한다, leaf-ordering)에 대해서, 기존 방법들을 살펴보고 새로운 방법을 제안하고, 각각의 장단점에 대한 분석을 하였다. 리프오더링 문제에 대한 자세한 설명은 2.2절에서 다루고 있다.

2. 연구 모델 및 리프오더링 문제의 정의

2.1 연구 모델

연구모델은 다음과 같다.

행렬 $D = \{d_{ij}, i=1,\ldots,m, j=1,\ldots,n\}$: DNA 마이크로어레이 실험 데이터로 구성된 $m \times n$행렬이다.

d_{ij} : i번째 유전자의 j번째 실험(어레이)에서의 발현도(expression level)를 나타낸다.

d_i : i번째 유전자의 모든 실험(어레이)에 대한 발현도를 담고 있는 행 벡터를 의미한다.

이진 트리 T : 계층적 클러스터링의 결과로 나온 이진 트리로써, m개의 리프 노드를 가진다.

l_k : 트리 T의 근노드를 왼쪽 편에 두고, 트리를 오른쪽 편으로 그렸을 때, 리프 노드들은 수직 방향으로 일렬을 지어 서게 된다. 이때 위에서부터, 차례대로 리프 노드들의 번호를 할당한다. l_k는 이런 방법으로 번호를 할당했을 때의 k번째($k=1,\ldots,m$) 리프 노드를 의미한다.

Z_T : 계층적 클러스터링의 결과물로 나온 트리 T의 리프 노드들(l_k)로 만들 수 있는 순서 리스트들의 집합을 의미한다. 유전자의 개수가 m개일 때, 순서 리스트의 가지 수 $|Z_T|$는 총 2^{m-1}개다[6,7].

2.2 리프오더링 문제의 정의

리프오더링 문제를 정형화하자면 다음과 같다.

첫 번째, 계층적 클러스터링을 거쳐서 나온 이진 트리에 대해서 트리 T의 모양을 변화시키지 않아야 한다. 단순히 노드들을 부모 노드를 중심으로 위아래만 바꾸어 줄 수 있다.

두 번째, 2^{m-1}개의 순서 리스트들 중에서, 인접한 리프 노드들간의 거리(유사도의 역수)의 총합이 가장 낮은 순서 리스트를 찾는다.

여기서, Z_T들의 총 가지 수가 2^{m-1}이기 때문에 단순 검색의 방법으로 해결 할 수는 없다. 따라서, 여러 가지 휴리스틱한 방법들이 제안되었고[3,4,5], 최근에는 동적 프로그래밍을 이용한 최적해 리프오더링 알고리즘이 개발되었다[6,7].

리프오더링과 생물학적인 의의 : 전산학적인 방법으로 최적의 리프오더링을 했을지라도 이것이 생물학적으로 가장 잘 된 리프오더링이라 말할 수는 없다. 단지, 리프오더링을 잘 하면, 생물학적인 연관성을 더 높일 수도 있다는 가정 하에서 리프오더링에 대한 연구는 진행되어 왔다고 볼 수 있다. 따라서, 앞으로 언급할 최적해 알고리즘이 생물학적인 결과가 가장 좋다고 보장할 수 없고, 무작위 방법으로 리프오더링하는 것이 생물학적인 결과가 가장 나쁘다고 말할 수도 없는 것이다. 다양한 생물학적인 데이터를 사용해서 직접 실험하면서, 그 효과를 평가해 보아야 한다.

3. 기존의 리프오더링 방법들

Eisen1 방법[3] : 첫 번째 방법은 휴리스틱한 방법의 대표

1) "이 논문은 2001년도 한국학술진흥재단의 지원에 의하여 연구되었음" (KRF-2001-041-E00265)

적인 것으로서, DNA 마이크로어레이 데이터의 계층적 클러스터링을 가장 먼저 시도하였던, Eisen의 방법이다. 각각의 행벡터 d_i에 대한 모든 어레이들의 평균 발현도를 구해서, 노드를 합병할 때, 발현도가 낮은 노드를 위쪽에 두고, 높은 노드를 아래쪽에 두는 방법이다.

Eisen2 방법[4] : 두 번째 방법도 역시 Eisen이 제안한 휴리스틱한 방법으로서, 그가 만든 소프트웨어인 Cluster의 사용자 설명서에서 제안한 것이다. 행렬 D에 대해서, 먼저 1차원 SOM(Self-Organizing Map)을 적용하여서, 클러스터링을 한다. 그리고, 결과로 나뉘어진 클러스터들에 대한 번호를 d_i에 기록한다. 그런 다음 다시 행렬 D에 대해서 계층적 클러스터링을 적용하되 두 노드를 합병하는 시점들마다 앞서 SOM을 이용해서 할당된 클러스터 번호를 참조하여서, 최대한 SOM에서 만들어진 리프 노드들의 순서가 유지되도록 하는 방법이다.

Alon 방법[5] : 세 번째 방법은 Alon이 제안한 휴리스틱한 방법이다. Alon은 [5]에서 확률적인 접근방식을 이용한 새로운 계층적 분할 클러스터링 방법을 제안하면서, 또 다른 휴리스틱한 리프오더링 방법을 고안하였다. 이 방법은 클러스터들을 분할해 갈 때, 새롭게 분할되어 가는 두 노드가 부모 노드의 형제 노드와 어느 쪽이 더 가까운 지를 확률적으로 계산해서, 가까운 쪽을 부모 노드의 형제 노드 쪽으로 배치하는 방법이다. 그러나, 이 방법은 Alon의 방법처럼, 분할 방식의 계층적 클러스터링 방법에서만 적용될 수 있는 제한성을 갖는다.

Random 방법 : 네 번째 방법으로 말할 수 있는 것은 무작위로 리프오더링하는 방법이다. 전산학적으로 좋은 결과가 나오기를 기대하기는 어렵지만, 다른 방법과의 비교를 위해서 실험해 보았다.

Optimal 방법[6,7] : 다섯 번째 방법은 MIT의 Bar-Joseph 등과 Waterloo대학의 Biedl 등에 의해서 연구된 최적해 리프오더링 방법이다. 이것은 동적 프로그래밍 기법을 적용해서, 주어진 클러스터링 트리 T에 대해서 최적해를 구하는 리프오더링 방법이다. 처음 개발되었을 때에는 $O(n^4)$알고리즘이었지만[6], 그 다음 논문에서는 $O(n^3)$알고리즘을 연구 개발하였다[7]. 그리고, 몇 가지 속도 향상 기법을 적용하여 검색 영역을 축소함으로써, 실제 수행 속도에 있어서 많은 발전을 이루었다. 그러나, 큰 데이터에 대한 수행 속도에서는 아직 느리다.

4. 제안하는 리프오더링 방법

제안하는 리프오더링 알고리즘은 각 유전자 벡터 d_i의 원소 중에서 가장 발현도가 큰 값의 j 인덱스(실험번호 또는 어레이 번호)를 참조해서 계층적 클러스터링을 하는 방법이다. 구체적인 알고리즘은 아래에 기술되어 있다. 아래의 알고리즘에서 r_i는 각 유전자 벡터 d_i를 클러스터링할 때, 순서를 정하기 위한 참조값이다.

for $i=1$ to m

$$r_i = \arg \max_j \{d_{ij}\}$$

위와 같은 과정에 의해서 r_i를 모두 계산한 후, 실제로 계층적 클러스터링을 할 때에는 참조값 r_i의 값이 낮은 노드를 위쪽 노드로 올린다.

따라서, 결론적으로 제안하는 리프오더링 알고리즘의 적용을 통해서 얻어지는 결과는

① 제안하는 리프오더링 알고리즘 적용 전과 동일한 구조의 결과 트리가 만들어진다.

② 리프 노드들의 배열 순서만 바뀌어진 상태가 된다.

제안하는 방법에서, 유전자 벡터 d_i 내에서 발현도가 가장 높은 실험(어레이)을 찾아서 그것을 기준으로 리프오더링을 한 이유는, 가장 발현도가 높은 실험(어레이)이 생물학적으로 그 유전자의 특성을 잘 알려주는 요인으로 생각하는 것도 의미가 있기 때문이다.

5. 리프오더링 방법들의 실험

여기서, 비교 분석하고자 하는 리프오더링 방법은 Random 방법과 Eisen1 방법, Optimal 방법, 그리고, 이 논문에서 제안한 방법이다.

실험 방법 : Eisen은 DNA 마이크로어레이에 대한 계층적 클러스터링을 Cluster라는 소프트웨어로 구현하였고, 이것은 클러스터링에서 가장 흔히 사용되는 소프트웨어 중의 하나가 되었다. 리프오더링 방법들의 실험에서는 이 Cluster 소프트웨어가 사용된다[4]. Random 방법, Eisen1 방법, Optimal 방법은 Cluster 소프트웨어를 사용해서 나온 결과 트리에 대해서 리프오더링 알고리즘을 적용하였다(방식1). 그리고, 제안한 알고리즘은 Cluster의 입력파일에 먼저 적용하고 Cluster 소프트웨어를 사용하였다(방식2). Cluster의 입력 파일 양식에는 이런 용도로 쓰일 수 있도록 GORDER라는 필드가 있다[4]. Eisen2 방법도 방식2를 이용하도록 되어 있다. 방식1과 방식2의 결과 트리는 같은 구조를 가진다.

실험 데이터 : 실험 데이터는 표 1과 같은 총10가지의 데이터를 사용하였다.

1),2)는 난수를 발생시켜 만든 후, Cluster 소프트웨어를 이용해서 정규화와 로그변환을 한 데이터들이다. 3)~7)은 Spellman[8]의 효모 데이터들이다. 5)는 [8]의 논문에서 나온 총 6178개의 데이터이고, 3)은 이중에서 Spellman이 세포주기(Cell Cycle)와 관련된 800개의 유전자를 뽑은 것이다. Bar-Joseph은 이것으로 생물학적인 분석을 하였다[6]. 4)는 그 중에서 cdc15에 대한 어레이만을 뽑아낸 데이터이다. 6)은 5)에서 20%이상의 데이터가 없는(missing) 필드를 가지고 있는 유전자들과 적어도 로그 비율로 2를 넘는 발현도가 한 번도 나오지 않은 유전자들에 대해서는 필터링하고 난 뒤에 남은 523개를 모아놓은 데이터 집합이다. 8)은 Alon[5]이 사용한 결장암(colon cancer)환자들에 대한 데이터 집합이다. 9)는 Golub등[9]이 AML과 ALL 환자들에 대한 테스트 데이터 집합이고, 10)은 트레이닝 데이터 집합이다.

표 1 실험에 사용된 데이터들

	데이터 이름	유전자 수	어레이 수	비고
1	r500_20	500	20	랜덤 데이터1
2	r800_20	800	20	랜덤 데이터2
3	Spellman799	800	82	효모 데이터1
4	SpellmanCDC	800	27	효모 데이터2
5	SpellmanAll	6178	82	효모 데이터3
6	Spellman523	523	82	효모 데이터4
7	SpellmanFun1033	1033	82	효모 데이터5
8	Alon2000	2000	62	결장암 데이터
9	GolubA	7129	34	AML/ALL 데이터1
10	GolubB	7129	38	AML/ALL 데이터2

6. 결과 분석

전산학적인 분석 : 알고리즘별로 리프 노드간의 거리 총합을 구한 것이 표 2에 나와있다. Optimal 방법이 가장 좋은 결과를 내었고, Random 방법이 좋지 않은 결과를 내었다. Eisen1 방

법과 본 논문에서 제안한 방법은 거의 비슷한 결과를 보여 준다. 그러나, Optimal 방법의 결과와는 다소 거리가 있다.

수행 시간 분석 : 표 3은 제안한 방법을 최적해 알고리즘의 수행속도와 비교한 것이다. 6000개 이상의 유전자를 가진 데이터에 대해서는 확연한 수행속도 차이가 있는 것을 볼 수 있다. 여기 표에는 나와 있지 않지만, Eisen1 방법과는 대략적으로 비슷한 수행속도를 보였다.

생물학적인 분석 : 생물학적인 분석은 [6]의 논문에 나와 있는 방식대로 리프오더링이 효모의 세포주기와 얼마나 연관성이 있는지를 보여주었다. 이 결과는 그림 1에 나타나 있다. 모두 다 같은 결과의 트리를 가지고 리프오더링을 한 것인데, 리프오더링 방법에 따라 많은 차이가 보인다. G2/M과 M/G1에 대해서는 제안한 방법이 유전자를 가장 잘 모아주는 것을 볼 수 있다. 반면에 Optimal 방법은 S/G2에 대해서 가장 잘 모아주는 것을 보여준다. 따라서, 전산학적인 최적해가 생물학적인 최적해가 되는 것이 아니라는 것을 볼 수 있고, 제안한 방법도 어느 정도의 의의를 가진 방법이라고 할 수 있다.

7. 결론 및 향후 연구 방향

본 논문에서는 계속적인 DNA 마이크로어레이 클러스터링 기술의 연구를 위한 한 과정으로써, 계층적 클러스터링 방법에 대한 여러 가지 리프오더링 방법들을 살펴보며 비교하였다. 새로운 휴리스틱 알고리즘을 제안해서 생물학적인 결과 분석도 보여주었다. 앞으로 여러 가지 리프오더링 방법들에 대한 다양한 생물 데이터를 이용한 실험이 필요하리라고 본다. 또한, 또 다른 리프오더링 방법이 더 연구될 수 있으리라고 본다. 이 논문에 대한 보조자료는 다음 URL에 존재한다.

http://alg.cse.cau.ac.kr/supplement/02kiss1

표 3 여러 가지 리프오더링 알고리즘별 리프 노드간의 거리 총합 비교

방법 / 데이터	Random	Eisen1	제안한 알고리즘	Optimal
r500_20	225.89	225.90	222.81	184.74
r800_20	344.94	342.33	343.80	281.21
Spellman799	330.13	334.12	333.61	289.75
SpellmanCDC	260.57	256.05	256.84	216.09
SpellmanAll	3032.03	3000.63	3010.27	2694.02
Spellman523	242.75	241.31	239.56	210.31
SpellmanFun1033	568.69	558.74	566.93	498.92
Alon2000	786.94	777.49	779.02	680.12
GolubA	2883.54	2882.56	2857.82	2482.62
GolubB	3029.63	3025.10	3005.77	2626.75

표 2 최적해 알고리즘과 제안한 알고리즘의 수행시간 비교 (dual Pentium III-933MHz, 1GB RAM)

방법 / 데이터	(A) 제안한 알고리즘	(B) Optimal	(A/B) 속도 비교
r150_20	0.001	0.05	2.00%
r500_20	0.004	0.33	1.21%
r800_20	0.009	0.88	1.02%
Spellman800	0.024	2.25	1.07%
SpellmanCDC	0.009	0.77	1.17%
SpellmanAll	0.205	151.04	0.14%
Spellman523	0.016	0.99	1.62%
SpellmanFun1033	0.037	4.12	0.90%
Alon2000	0.046	26.92	0.17%
GolubA	0.137	145.88	0.09%
GolubB	0.154	149.23	0.10%

참고 문헌

[1]여상수, 김성권,"DNA 마이크로어레이 데이터 클러스터링 알고리즘의 연구 동향", 한국정보과학회 컴퓨터이론연구회지, 제12권 1호, pp.2-11, 2001년10월

[2]R. Shamir and R. Sharan. "Algorithmic approaches to clustering gene expression data", In Current Topics in Computational Biology. MIT Press, submitted.

[3]M. Eisen et al., "Cluster analysis and display of genome-wide expression patterns", Proc. of Natl. Acad. Sci., 95:14863-14867, 1998.

[4]http://rana.lbl.gov/manuals/ClusterTreeView.pdf

[5]U. Alon et al., "Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays", Proc. Natl. Acad. Sci., 96:6745-6750, 1999.

[6]Z. Bar-Joseph et al., "Fast optimal leaf ordering for hierarchical clustering", In Proceedings of ISMB 2001.

[7]http://monod.uwaterloo.ca/supplements/01expr/art.pdf

[8]P.T. Spellman et al. "Comprehensive identification of cell cycle-regulated genes of the yeast Saccharomyces cerevisiae by microarray hybridization", Mol Biol Cell 9:3273-97.1998.

[9]T.R. Golub et al., "Molecular classification of cancer: class discovery and class prediction by gene expression monitoring", Science. 286:531-537, 1999.

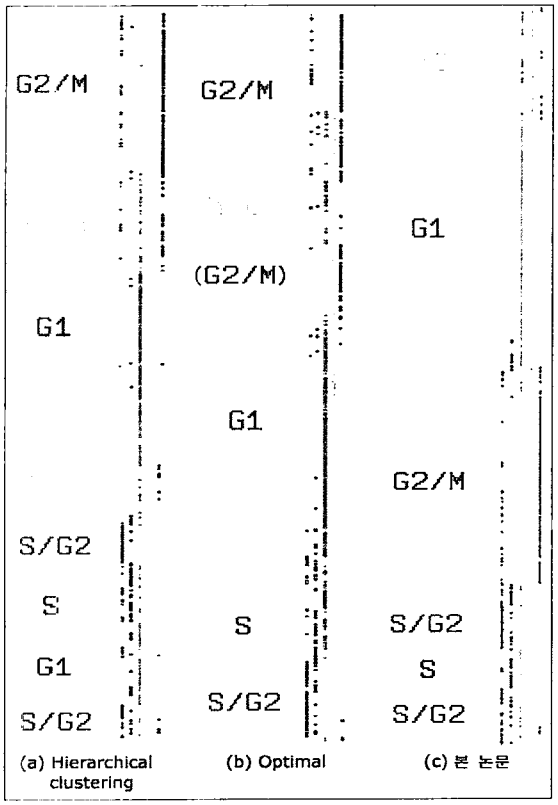

그림 1 리프오더링을 적용하지 않은 것(a)과 Optimal 방법(b)과 제안한 방법(c)의 생물학적인 분석