계산적 계통분류학

부산대학교 조환규* · 최정현

1. 서 론

생물학에서 생물의 각 단위(taxonomical unit), 즉 개별 종이나 군집(population)의 계통분류를 결정하는 일은 대단히 중요한 문제라고 한다[1,2]. 생물학 연구자들에게 진화의 과정을 정확히 역추적하여 그 계통분류를 추정하는 일은 계산학자에게 어떤 선형시간의 optimal 알고리즘을 개발하는 것만큼이나 중요한 문제이다. 따라서 이번 글의 목적은 생물종을 분류하는 계통발생학(phylogenetics)이라는 분과 학문을 계산학적인 관점으로 살펴보는 것이지 생물체 분류 그 자체의 의미에 대하여 설명하는 것은 아니다. 즉 생물체의 진화적 과정을 역추적하여 그 갈래를 결정하는 것을 형식적인 1한 계산문제로 바꾸었을 때 그것이 얼마나 어려운 계산문제인지, 그리고 그 어려움에 대처하는 다양한 휴리스틱은 어떤 것인지를 살펴보는 것이 이 글의 목적이다.

분자 생물학이 크게 발전하지 않았을 경우에는 대부분 형태적(morphological) 분류법이 사용되었다. 예를 들어 사자와 개의 형상(주둥이나 발의 모양, 털의 얼룩모양, 생활습성 등)을 주된 판정 기준으로 삼았다. 물론 그 분류법으로 정해진 여러 종이 있지만 미생물의 경우에는 그 형상의 특징을 추출하기가 상당히 힘들므로 형상만 가지고 분류하는 데에는 한계가 있다. 따라서 진화적 과정이 고스란히 유전자의 염기서열에 기록되어 있다고 한다면 유전자들간의 관계를 이용해서 분류를 시도하는 것이 가장 합리적인 방편이 될 것이다. 단 그 분류의 과정은 tree 형식으로 된다고 가정한다. 따라서 현재의 생물체(present day object)의 상태를 입력으로 사용하여 진화의 갈래를 계산해서 그 결과를 계통도(phylogenetic tree)로 그려내는 것이 계산적 계통발생학의 목표라고 할 수 있다. 아래 그림 1에 사람(human)이 포함된 계통도가 있다.

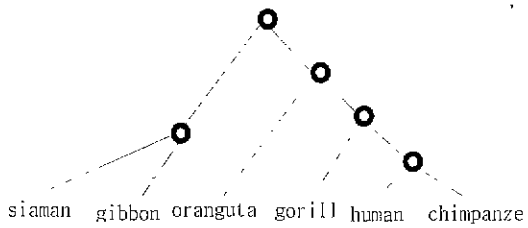


그림 1 계통도(phylogenetic tree)의 한 예

그런데 계통발생학의 기본적인 가정은 하나의 생물체에서 진화와 변이를 거쳐서 지금의 모든 생물체가 만들어 졌다는 것이다. 물론 이것은 가정이지만 많은 실험결과들이 이러한 사실을 나름대로 잘 반증해주고 있다. 하지만 최근의 분자생물학의 연구 결과에 의하면 하나의 원생물체만을 가정하는 것(central dogma)보다는 복수개의 원생물체를 가정해서 분류과정을 설명하는 것이 더 합당하다는 연구결과가 대두되고 있어 흥미를 더하고 있다.

분류학에서 과학자들이 관심을 가지고 있는 것은 갈래의 모양(topology)과 각 taxonomical unit들 사이의 진화적 거리(evolutionary distance)이다. 진화적 거리를 계산해 봄으로서 어떤 종이 한 종에서 분화되어 나온 상대적인 시

* 정회원

간을 가름해 볼 수 있다. 또한 유사종들 사이에서 시발 종(root)이 있는 지를 살펴보는 문제도 매우 중요한 문제이다. 물론 분자생물학에 근거한 계산적 분류방법 외에도 여러 가지 분류 방식이 있으며, 어떤 것이 가장 좋은 방식인지를 판별할 정량적인 방법은 없다.

계통도를 결정하기 위하여 필요한 입력 자료의 형식에는 크게 두 가지가 있다. 하나는 개체의 각 특성을 이산적(discrete)인 상태로 표시한 character state matrix와 모든 개체 쌍들간의 진화적 거리(evolutionary distance)를 나타낸 distance matrix이다. 따라서 계통도를 구상하기 위하여 준비된 입력이 무엇인가에 따라서 실제 트리를 구성하는 방법도 다르다[1].

2. Perfect Binary Phylogeny Problem

일단 입력이 다음과 같은 character state matrix로 되어 있다고 생각해보자.

표 1 어떤 5개 개체(A,B,C,D,E)의 character matrix

	Character					
object	c1	c2	c3	c4	c5	c6
A	0	0	0	1	1	0
B	1	1	0	0	0	0
C	0	0	0	1	1	1
D	1	0	1	0	0	0
E	0	0	0	1	0	0

위의 표에서 나타난 바와 같이 개체는 모두 5개이며 각 개체는 6개의 상태를 가진다 위의 표에서 상태는 0/1의 이진상태인데 어떤 특성이 있는지 없는 지를 나타낸다고 보면 된다. 예를 들어 c1을 "꼬리가 있다"라는 표현으로 이해하면 된다. 문제는 이 5개의 개체를 입력으로 갈래를 나누고자 한다. 여기에서 perfect phylogeny라는 특성이 정의해보면 다음과 같다. n 개의 개체를 이용해서 rooted tree를 만들고 각 leaf node마다 개체를 배열한다고 해보자. 이 경우 중간의 internal edge에 대하여 각 character state를 적절히 중복 없이 배치하여 그 에지를 bridge로 하는 두 개의 component에 개체들의 character state의 값이 완전히 분리되도록 배치할 수 있으면 "그 character matrix에는 perfect phylogeny가 존재한다"라고 말한다. 예를 들어 위의 character matrix를 다음과 같은 한 계통도로 만들었다고 해보자.

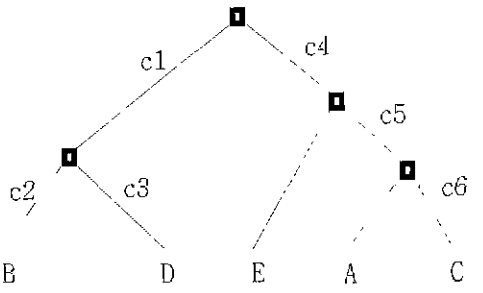


그림 2 표 1 자료에 근거하여 만든 한 계통도 (phylogenetic tree)

이 경우 c1에 해당되는 에지를 중심으로 B,D는 모두 c1=1이며, E,A,C는 모두 c1=0이다. 동시에 c4에 대하여 살펴보더라도 c4 에지를 중심으로 양쪽으로 나누어져 있는 종들이 c4로 완전히 양분된다 c5에 대하여 살펴보면 B,D,E는 모두 c5=0이며 나머지는 모두 1로 표시된다. 주어진 character matrix를 보고서 perfect phylogeny가 있는지를 판별하는(yes/no question) 문제가 흥미로운 문제이다. brute force한 방식은 모든 가능한 tree topology를 고려하는 것인데 불행히도 n개의 노드로 구성되는 vertex labelled tree의 수는 n!개 이상이므로 이 방식은 사용 불가하다. 문제를 좀 좁혀서 각 character state의 값이 0또는 1일 경우라면 이 perfect phylogeny 문제는 O(n*m)에 해결된다. 여기에서 n은 개체의 수이며 m은 한 개체가 가지는 character 의 종류이다. 아래 정의에 의해서 matrix의 compatability가 정의된다.

정의 1: 어떤 binary matrix M이 perfect phylogeny를 가지는 필요충분조건은 임의의 두 개의 column이 서로 disjoint하거나 한 column이 다른 column에 완전히 포함되는(contain) 경우이다.

물론 위 정의대로 단순한 알고리즘을 구성하면

$O(n*m*m)$이 되지만 몇 가지 중복되는 경우를 제외하는 기법을 사용하면 $O(n*m)$에 해결할 수 있다[1].

그러나 실제 상황에서 character state matrix의 값이 binary일 경우는 거의 없다고 보아도 무방할 것이다. 동시에 그것이 또한 compatible matrix일 가능성도 매우 희박하다 따라서 이론적인 관점에서 위의 문제를 좀 더 완화시키면 두 개의 character이면서 각 state는 몇 개의 값을 허용하는 문제를 고려해 볼 수 있다. 이것은 주어진 matrix에서 state intersection graph(SIG)를 만드는 것으로부터 시작한다 임의의 character matrix에서 SIG는 다음과 같이 만든다. 먼저 각 character state의 값을 모두 vertex로 지정하고 한 개체에 나타나는 모든 character들을 clique로 연결한다. 즉 자신들끼리 모두 edge가 있도록 연결한다. 이렇게 두 개의 character state만 가진 matrix에서 생성된 SIG가 acychc이면 그 해당되는 matrix는 perfect phylogeny를 가짐이 밝혀져 있다. 따라서 이것은 linear time에 구할 수 있다.

3. Parsimony and Compatability

앞서 설명한 Perfect Binary Phylogeny 문제가 polynomial time에 해결됨을 알 수 있었는데 이것은 0과 1의 상태가 두 개이기에 가능한 것이다. 만일 주어진 matrix의 character state의 값이 복수 개라면 이러한 문제는 일반적인 perfect phylogeny problem인데 보통 perfect phylogeny with unordered character라고 불리는 이러한 문제는 이미 NP-complete임이 밝혀져 있다. 따라서 perfect phylogeny라는 decision problem을 optimization problem으로 전환하여 살펴보는 것이 바로 parsimony라는 개념이다. 사실 진화의 과정이 단선적이 아님은 여러 연구에서 밝혀진 바도 있거니와 그것이 측정에서 완벽한 계통분류를 만들어 낸다는 것은 거의 기대하기 힘든다. 따라서 우리는 가능하면 perfect phylogeny를 이루는 가장 큰 character들의 subset을 구하고 이를 기준으로 해서 현실적인 대안책을 마련할 수 있다.

따라서 이 문제는 주어진 matrix에서 compatible matrix가 될 수 있는 가장 많은 수의 column을 선택하는 문제로 귀착될 수 있다. 그리고 이 문제는 다시 어떤 두 character가 compatible하면 에지를 가지는 그래프로 변환하였을 때에 그 그래프에서 maximal clique를 찾아내는 문제와 동등하므로 쉽게 NP-complete임이 증명된다. 이 때문에 perfect phylogeny를 구하는 문제는 clique finding에 관한 여러 가지 heuristic이나 approximation algorithm으로 그 근사해를 구할 수 있다

4. Distance Matrix method

지금까지와는 달리 임의의 두 개체간의 진화적 거리를 어떠한 방법으로든 계산할 수 있다면 그것으로도 계통분류를 유추해 볼 수 있다. 예를 들어 우리가 어떤 6개의 종들간에 어떤 한 방법을 사용하여 그 상대적인 진화적 거리를 구했다고 가정 해보자(아래 표 2를 참조하자).

표 2 6개 개체들간의 진화적 거리 행렬

	A	B	C	D	E	F	G
A		63	94	111	67	23	107
B			79	96	16	58	92
C				47	83	89	43
D					100	106	20
E						62	96
F							102

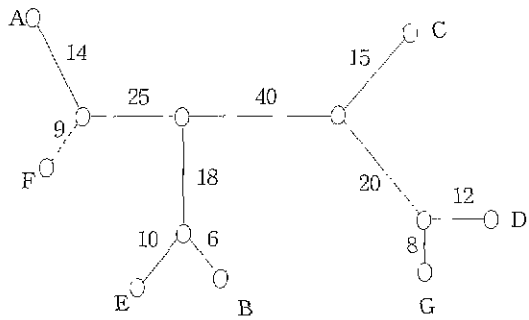


그림 3 표 2의 거리제한 조건을 만족시키는 tree

다행히 위와 같은 거리 행렬(distance matrix)이라면 다음과 같은 모양의 트리로서 그려낼 수 있다. 즉 두 종간의 거리는 트리에서의 경로의 길이로 표시된다. 그러나 짐작하셨다시피 임의로 주어진 symmetric matrix에서 각 종들간

의 거리를 유지하도록(preserve) 트리를 그리는 것은 불가능하다. 따라서 주어진 거리행렬이 트리로 그려질 수 있는 경우는 아주 특수한 additive matrix인 경우뿐이다.

정의 2: 어떤 matric space O가 additive(가산적)일 경우의 필요 충분 조건은 다음과 같다. O에 속한 임의 4개의 원소 i,j,k,l에 대하여 그 거리를 d(,)로 나타낸다고 한다면 다음의 식을 항상 만족시키는 경우이다. $d(i,j) + d(k,l) = d(i,k) + d(j,l) \geq d(i.l) + d(j,k)$.//

어떤 distance matrix가 additive하다면 그것으로 tree topology와 각 에지들의 길이를 구하는 문제는 비교적 쉽다[1]. 그런데 문제는 각 개체들간의 유전적 거리를 측정하는 방식도 상당히 다양하며 그렇게 측정된 거리행렬이 우연히 가산적 거리행렬이 될 경우는 거의 없다고 보아야 할 것이다. 왜냐하면 대부분의 유전적 거리측정 함수는 대개 DNA sequence나 Protein sequence의 쌍이나 multialignment를 통하여 구성되는데 이 과정에서 그 측정 metric은 전혀 additive한 특성을 고려하지 않는다. 예를 들어 두 프로틴(protein) 순서들간의 진화적 거리를 측정하는 PAM metric이나, Nakajima-Nie measure, 또는 두 DNA 스트링의 진화적 거리를 측정하는 Jukes-Cantor model의 distance metric은 additivity와는 아무런 상관이 없다.

따라서 현실적인 문제를 고려해서 거리행렬을 가능한 가산적 행렬로 만들고자 하는 relaxation의 한 방법이 제시되었는데 그것은 두 개체간의 거리에 upper, lower bound을 주는 것이다. 즉 $\min_{i,j} \leq d(i,j) \leq \max_{i,j}$ 의 min 값과 max 값을 모든 i,j쌍에 지정하여 그 한계 내에서 additive distance matrix가 되도록 만드는 것이다. 이렇게 제한 범위 내에서 additive distance matric을 유지하는 것을 우리는 그 거리행렬이 ultrametric을 가진다고 말을 한다. 따라서 ultra matric인지를 검사하기 위해서는 각 쌍들간의 거리의 min값을 가진 행렬과 max값을 가진 행렬 두 가지가 주어져야 한다. 이러한 두 개의 거리행렬이 ultrametric인 경우, 그것으로부터 실제의 계통도를 구성하는 과정은 minimal spanning tree로의 변형과 복잡한 과정을 거쳐서 O(n*n) 시간 내에 판별할 수 있다.

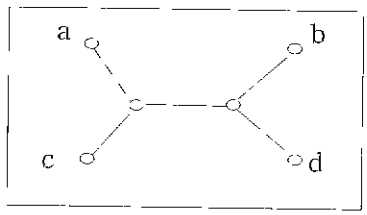
다른 방식으로 계통도의 성능을 측정할 수 있는데 그것은 두 종들간의 진화적 거리 $D_{i,j}$ 와 그들이 어떤 계통도 T로 표현되었을 경우, 모든 종들의 쌍 i,j 사이의 트리 T에서의 거리($d_{i,j}$)의 차의 제곱의 합이 얼마나 적은지를 계산해 봄으로서 예상이 가능하다. 그러한 measure를 Least Square Method에 의한 측정이라고 말하며 그 식 SSQ(T)는 다음과 같다.

$$SSQ(T) = \sum_{i=1}^{n} \sum_{j \neq i} w_{i,j} (D_{i,j} - d_{i,j})^2$$

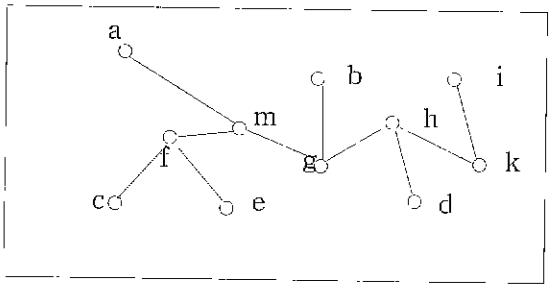
위 식에서 $w_{i,j}$는 weight 변수이다. 그런데 주어진 거리 행렬에서 이러한 SSQ measure를 최소화하는 트리를 구성하는 것은 NP-complete임이 밝혀졌다[8]. 따라서 여러 가지 heuristic algorithm이 개발되었는데 가장 대표적인 알고리즘은 UPGMA(Unweighted Pair Group Method with Arithmetic)과 Neighbor Joining, 그리고 통계적인 방법을 이용한 Maximum Likelihood를 이용한 방식이 있으며 최근에 각 방법을 변형하거나 조합한 새로운 유형의 heuristic도 많이 소개되고 있다.

5. Quartet Method

지금까지의 방식과는 좀 다른 방법으로 계통분류를 유추해보는 방법론도 있다. 이것은 n개의 종들간에 모든 4개 종들간의 계통분류를 구성한다. 이렇게 구성된 4개 종의 계통분류를 quartet라고 부르는데 그 종류는 모두 O(n^4)개가 존재한다. 문제는 n개의 노드로 구성된 super tree를 구하되 가능하면 그전에 만든 quartet가 가장 많이 super tree와 homeomorphic하도록 super tree를 구성하는 것이다. 어떤 quartet가 super tree와 homeomorphic하다고 하는 것은 그 quartet에 포함된 각 노드에 이르는 경로외의 모든 경로를 제거해서 만든 트리가 quartet와 isomorphic한 것을 말한다. 그림 4(a)는 어떤 4개의 개체로 구성된 한 quartet를 보여주고 있다. 그리고 그림 4(b)는 11개의 노드로 구성된 한 계통도인데 이 트리는 그림 4(a)에 나타난 quartet를 포함하고 있다.



(a) Quartet



(b) 어떤 Phylogenetic tree

그림 4 quartet 방식에 의한 phylo tree 구성

따라서 문제는 그림 4(a)에 나타난 형태의 ${}_nC_4$ 개의 quartet를 모두 포함할 수 있는 트리를 구성하는 것이다. 그런데 임의로 주어진 ${}_nC_4$ 개의 quartet를 모두 포함하는 트리를 구성할 수 없으므로 결국 optimization problem이 된다. 그리고 가장 많은 수의 quartet를 homeomorphic하게 포함하는 tree를 구성하는 문제는 NP-complete임이 밝혀져 있다. 따라서 적절한 휴리스틱을 개발할 필요가 있는데 semidefinite programming 방법론과 geometric algorithm을 이용해서 상당한 정도의 결과를 보장해주는 휴리스틱 알고리즘이 개발되어 있다[3].

6. 최근 연구 동향

한 가지 아주 흥미로운 연구는 현재의 computer virus에 관한 계통도를 구성하는 것이다[6]. 현재 대략 3-4일에 하나씩 새로운 컴퓨터 바이러스가 만들어지고 있다고 한다. 바이러스를 만드는 과정은 대개 이미 만들어진 기존의 바이러스 프로그램에 약간의 수정을 가해서 새로운 종을 만들어 내므로 그러한 바이러스 류에도 계통도가 존재한다. 그리고 공학적인 이유에서 살펴보면 전체 파일 시스템에서 바이러스 프로그램이 있는지를 살펴보는데 이 트리가 아주 중요한 역할을 수행한다. 대략 알려진 3000여종의 바이러스의 특징부분(대략 20바이트 내외)을 수만 가지의 파일에 각각 비교하는 것은 매우 많은 시간을 요하므로 반드시 각 개체별 공통의 스트링을 사용해서 비교를 해야만 적절한 시간 내에 검사가 끝난다. 참고문헌[6]에는 이에 관련된 많은 계산문제와 그 복잡도가 나열되어 있다.

여러 개체들 사이의 계통도를 구성하면 항상 불일치가 생길 수 있으므로 트리라는 제한조건을 풀어서 cycle이 생길 수 있도록 계통도를 구성하는 것에 관한 연구결과도 있다[9]. 그리고 트리를 보기 좋게 그려내는(drawing) 문제도 고려해 볼 수 있다. "보기 좋은" 정도는 여러 가지로 측정할 수 있는데 edge crossing이라든지, 각 leaf node들이 분산해있는 정도(uniformity) 등을 measure로 해서 새로운 graph drawing 문제로도 연구해 볼 수 있다. 특히 100여 개체 이상의 자료로부터 계통도를 그려내는 문제는 또 다른 연구문제를 생각해 볼 수 있다.

본 논문의 저자들이 만든 프로그램으로 PhyloVIS라는 도구가 있다[11]. 이 프로그램은 기존의 여러 가지 multi-alignment 도구로부터 만들어진 다양한 distance matrix 출력물을 통합적으로 읽어서 postscript 파일형식의 계통도를 그려주는 기능이 있다. 이 프로그램은 상용 프로그램인 MEGA, PHYLIP, Clustal-W, NEXUS, PAUP 등의 프로그램 출력형식을 모두 지원하며, 각종 다양한 형식의 트리(radial, rooted, unrooted 등)를 사용자가 제시한 형식에 맡게 그려주는 기능이 있다. PhyloVIS로 그려진 한 tree의 모양과 전체 인터페이스가 아래 그림 5에 있다.

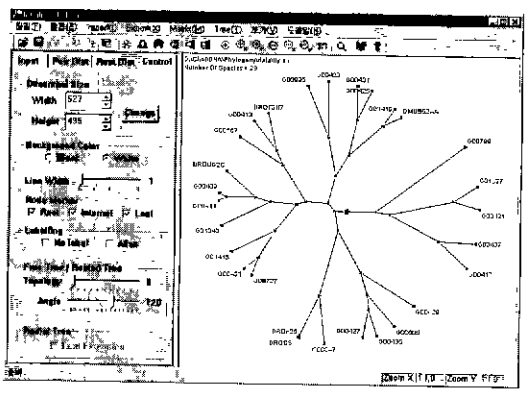


그림 5 PhyloVis로 그려진 계통도[11]

참고문헌

[1] Setubal and Meidans, *Introduction to Computational Molecular Biology*, PWS Publishing, 1997.

[2] Dan Gusfield, *Algorithms on Strings, Trees, and Sequences*, Cambridge Univ. Press, 1997.

[3] Dan Pelleg, "Algorithms for Constructing Phylogenies from Quartet," MS thesis, Dept. of Comp. Sci., Israel Institute of Technology, Apr., 1998.

[4] Tao Jiang et al., "Orchestrating Quartets: Approximation and Data Correction," TR, Dept. of Computer Science, Univ. of Waterloo, 1998.

[5] Ron Shamir, Lecture Notes: Algorithms for Molecular Biology(Lecture 9) , CS dept., Tel Aviv University, 1999.

[6] Leslie Ann Goldberg, "Constructing Computer Virus Phylogenies," J. Algorithms, Vol.26, pp.188-208, 1998.

[7] K. Atteson, "The Performance of Neighbor-Joining Methods of Phylogenetic Reconstruction," Algorithmica, Vol.25, pp.251-278, 1999.

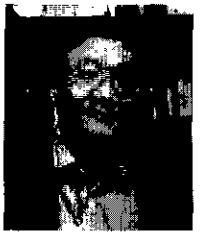
[8] W.H.E.Day, "Computational complexity of inferring phylogenies from dissimilarity matrices," Bulletin of Mathematical Biology, vol.49, pp.461-467, 1986.

[9] D.Huson, "SplitsTree : Analyzing and visualizing evolutionary data," BIOINFORMATICS, Vol. 14, No.1, pp.68-73, 1998.

[10] S.K. Kannan and T.J. Warnow, "Inferring evolutionary from DNA sequences," SIAM J. on Computing, Vol.23, No.4, pp.713-737, Aug., 1994.

[11] J.H. Choi, H.Y. Jung, H.S. Kim, and H.G. Cho, "PhyloVIS: A phylogenetic tree viewing system," Technical report (submitted), 2000.

조 환 규



1980~1984 서울대학교 계산통계학과 학사

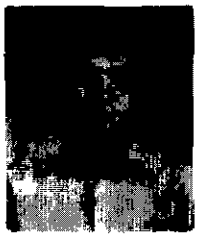
1984~1990 한국과학기술원 전산학과 석사,박사

1990~현재 부산대학교 전자계산학과 조교수 공과대학 정보컴퓨터공학부 교수(현)

관심분야:알고리즘 이론, 컴퓨터 그래픽스

E-mail:hgcho@hyowon.pusan.ac.kr

최 정 현



1986~1995 부산대학교 물리학과 학사

1994~1998 otlas communications

1998~2000 부산대학교 전산학과 석사

관심분야:알고리즘 이론, 컴퓨터 그래픽스, 생명정보학

E-mail:jhchoi@pearl.cs.pusan.ac.kr