DOI: 10.3745/KIPSTA.2009.16-A.3.143

계층적 정렬쌍 가시화를 이용한 유전자 클러스터 탐색 알고리즘

진 희 정[†] · 박 수 현[††] · 조 환 규[†††]

요 약

최근 생물정보학 분야의 연구는 하나하나의 유전자를 연구하던 예전의 방법에서 유전자들 간의 관계를 알아보는 연구들로 변해가고 있다. 이러한 유전자들 간의 연구 중 하나가 유전자 팀(gene team)을 연구하는 것이다. 유전자 팀이란 몇몇 염색체들 사이의 유전자들이 보존되어 있는 것을 말하며, 닫힌 영역 안에 보존되어 있는 유전자들의 집합으로 볼 수 있다. 이들은 진화과정을 거치면서, 유전자 팀 내의 유전자들의 위치나 그 종류가 변한다. 이러한 유전자 팀을 찾기 위해 많은 연구들이 이루어져왔다. 본 논문은 생물정보학 분야에서 많이 사용되는 계층적 클러스터링(hierarchical clustering)방법을 변형하여 전체 유전체(whole genome) 쌍내에서의 의미 있는 영역을 찾고, 영역 내에서 gene team을 찾을 수 있는 방법을 소개한다. 본 연구 방법을 이용하면, 복잡한 구조의 두 유전체 사이의 연관 유전자들이나 유사 영역들의 맵(map)을 단계별로 간략화 하여 나타낼 수 있다.

키워드 : 생물정보학, 유전자 클러스터, 유전자 팀, 가시화, 간략화

A Gene Clustering Method with Hierarchical Visualization of Alignment Pairs

Jin hee-Jeong[†] · Park Su-Hyun[††] · Cho Hwan-Gue[†††]

ABSTRACT

One of the main issues in comparative genomics is to study chromosomal gene order in one or more related species. For this purpose, the whole genome alignment is usually applied to find the horizontal gene transfer, gene duplication, and gene loss between two related genomes. Also it is well known that the novel visualization tool with whole genome alignment is greatly useful for us to understand genome organization and evolution process. There are a lot of algorithms and visualization tools already proposed to find the "gene clusters" on genome alignments. But due to the huge size of whole genome, the previous visualization tools are not convenient to discover the relationship between two genomes. In this paper, we propose AlignScope, a novel visualization system for whole genome alignment, especially useful to find gene clusters between two aligned genomes. This AlignScope not only provides the simplified structure of genome alignment at any simplified level, but also helps us to find gene clusters. In experiment, we show the performance of AlignScope with several microbial genomes such as B. subtilis, B.halodurans, E. coli K12, and M. tuberculosis H37Rv, which have more than 5000 alignment pairs (matched DNA subsequence).

Keywords : Bioinformatics, Gene Cluster, Gene Team, Visualization, Simplification

1. 서 론

유전체에 포함되어 있는 유전자들은 진화 과정을 통하여 그 순서나 위치가 변하거나 삭제되기도 하고 새로운 유전자가 생겨나기도 한다. 이러한 진화과정을 거쳐 온 유전자들 중에서 종들 간에 순서나 방향이 유사한 유전자들이 존재하는데, 이러한 보존된 유전자들은 생명을 유지하는데 필요한 기능을 담당하고 있는 것들이 많다. 따라서 종들 간의 유사한 유전자들에 대한 연구가 진행되었으며, 최근에는 유사한 유전자들 하나하나를 연구하는 것이 아니라 특정 영역 내에서 함께 보존되어 온 유전자들의 집합을 찾아내는 연구들이 많이 진행되고 있다. 이러한 유전자들의 집합을 유전자 클러스터(gene cluster) 또는 유전자 팀(gene team)이라고 한다. 보존된 유전자 클러스터는 직간접적으로 기능 모듈을

※ 이 논문은 2006년도 정부재원(교육인적자원부 학술연구조성사업비)으로 한국학술진흥재단의 지원을 받아 연구되었음(KRF-2006-521-D00379).
† 정 회 원 : 한국한의학연구원 선임연구원
†† 정 회 원 : 삼성전자
††† 정 회 원 : 부산대학교 정보컴퓨터공학부 교수
논문접수 : 2008년 12월 4일
수 정 일 : 1차 2009년 2월 4일
심사완료 : 2009년 2월 23일

연구하는데 도움이 되며, 이들은 다른 유전자들에 비하여 가까이 위치하고 있다. 본 논문에서는 두 유전체의 정렬 쌍들을 이용하여 수정된 계층적 클러스터링 방법으로 유전자 팀을 찾고, 이를 가시화하는 툴인 AlignScope 시스템에 대해서 소개한다. 특히, 유전자 팀에 속한 유전자들의 보존성이나 순서들의 특징을 잘 표현할 수 있도록 정렬 복잡도(alignment complexity)라는 개념에 대해서 소개한다. 이를 통한 본 연구의 목표는 다음과 같다.

(1) 기존의 방법에서 찾을 수 없었던 두 유전체 사이의 유전자 팀을 찾는다.
(2) 두 유전체 사이의 유사 영역의 특징을 파악하기 쉽도록 간략화 가시화를 제공한다.

2. 관련 연구

사용할 수 있는 유전체 서열이 늘어남에 따라, 보존된 영역을 찾거나, 유전자 클러스터를 찾거나, 다른 종들 사이에 흥미로운 부분을 찾는 연구들이 가능하게 되었고, 유전자 클러스터를 찾는 문제 또한 그들의 연관 유전자(orthologous gene)들을 정의하고 그들을 함께 그룹으로 묶어주는 연구들이 진행되어 왔다. Overbeek[1]는 유전체들 사이의 유전자 클러스터들의 보존성에 기반을 두어 그룹화 하였다. Overbeek는 *run* 개념을 만들었는데, *run*은하나의 그룹 안에 있는 유전자들이 같은 방향으로 존재하며 유전자들 사이의 간격은 300bp 이하인 유전자들의 집합을 말한다. 하나의 *run*내의 유전자들을 닫혀있다(closed)고 말한다. (그림 1)은 Overbeek가 정의한 유전체 사이의 닫힌 유사쌍(pairs of close homolofs, PCHs)을 설명한다.

Ogata[2]는 대사 경로에서 그래프 이론을 바탕으로 유사성이 높은 지역을 찾아내었다. Ogata는 유사성이 높은 지역을 서로 연관한 클러스터(cluster)라고 하였다. Tatusov[3]는 계통도 분석, 클러스터 분석, 정렬 방법을 사용하여 양방향으로 연관된 유전자를 선택하고, 선택된 유전자들을 이용하여 COG(clusters of Orthologous Groups of Proteins) 데이터베이스를 구축하였다. Nicolas[4]는 몇몇의 유전체에서 연관된 유전자들을 찾은 다음, 비교하고 있는 유전체에서 특정 길이를 고려하여 함께 존재하는 연관된 유전자들을 찾아 유전자 팀(gene team)이라고 하였다. 이때, Nicolas는 각 유전자를 하나의 문자로 인식하고 이들의 집합을 유전체에서 유전자의 위치를 고려하여 서열로 나타내어 사용하였다.

지금까지 다양한 방법으로 유전자 팀을 찾으려는 노력들이 있어왔다. 하지만 기본적으로 모든 알고리즘들은 몇 개의 오류를 포함할 수는 있지만 각 유전체 간의 유전자 팀에서는 다른 유전자들의 삽입 등이 극히 적다는 것이다. 이러한 경우 예제 2.1과 같은 경우는 찾을 수 없다.

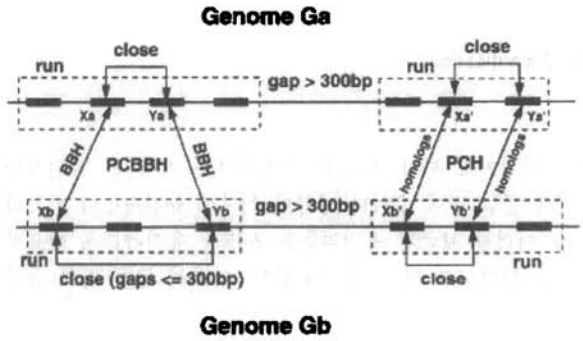
[예제 2.1] *두 유전체 G_a와 G_b가 있고, 각 유전체에 존재하는 유전자는 다음과 같다. *는 다른 유전자나 사이의 간격을 말한다.*

G_a의 유전자 위치 : $**,g_1,g_2,g_3,g_4,g_5,g_6,g_7,***$

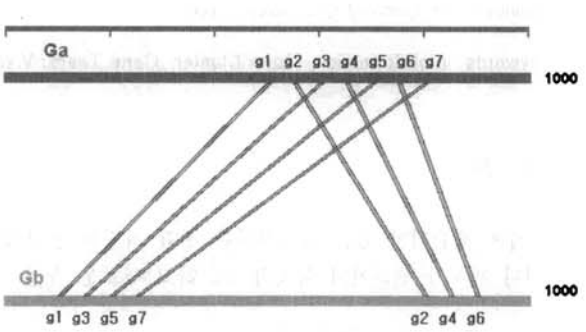
G_b의 유전자 위치 : $**,g_1,g_3,g_5,g_7,***,g_2,g_4,g_6,**$

예제 2.1과 같은 경우에는 G_b 유전체의 "$g_1,g_3,g_5,g_7(gc_1)$"과 "$g_2,g_4,g_6(gc_2)$"는 G_a의 "$g_1,g_2,g_3,g_4,g_5,g_6,g_7(gc_3)$"내에 모두 포함되지만, gc_1과 gc_3, gc_2와 gc_3를 각각 하나의 클러스터로 보기에는 G_a 유전체에서 많은 에러가 발생하며, gc_1과 gc_2를 포함하는 전체 영역과 gc_3를 하나의 클러스터로 하기에는 G_b 유전체에서 많은 에러가 발생한다. 따라서 이러한 경우 각각에 대한 유전자 팀은 찾을 수 없다. (그림 2)는 예제 2.1을 그림으로 나타낸 것이다.

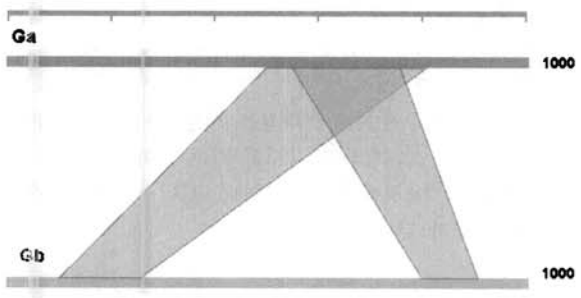
본 논문은 두 유전체의 전체 영역에서 유전자 팀을 포함한 의미 있는 영역을 찾아내는 방법론에 대해서 소개한다. 본 연구에서는 마이크로어레이 분석 시 많이 사용하는 계층적 클러스터링 방법론을 기반으로 하며, 이를 전체 유전체에 사용할 수 있도록 수정하여 사용한다. 본 연구에서 개발



(그림 1) PCHs(pairs of close homologs)의 설명 : 유전체 G_a와 G_b에 존재하는 각각의 *run*들과 그 *run*들 사이의 유사영역 PCHs를 나타낸다. PCHs 중 가장 유사도가 높은 영역을 PCBBH(pairs of close bidirectional best hits)로 정의한다 [1]



(그림 2) 유전자 팀을 찾는 예제 : 두 유전체 G_a와 G_b 사이에서의 유전자 팀을 찾기. G_a의 유전자들은 $**,g_1,g_2,g_3,g_4,g_5,g_6,g_7,***$, G_b의 유전자들은 $**,g_1,g_3,g_5,g_7,***,g_2,g_4,g_6,**$의 순서로 위치하고 있을 때, 기존의 방법으로는 두 유전체 사이에서 gene team을 찾을 수 없다



(그림 3) 예제 2.1에 대한 유전자 팀의 결과 : 본 연구에서는 계층적 클러스터링 방법론에 기반을 두어 전체 유전체에서의 유전자 팀들을 찾아낼 수 있다

(그림 4) E. coli K12와 M. tuberculosis H37Rv 간의 유사 유전자들 : E. coli K12와 M. tuberculosis H37Rv의 유사 영역(유전자 영역만을 고려)는 모두 1,487개이다. E. coli K12와 M. tuberculosis H37Rv 유전체의 길이가 아주 크고, 그 사이의 유사 영역의 수 또한 많기 때문에 하나의 화면에 가시화하면, 두 유전체 간의 특징을 파악하기 힘들다

한 방법론을 사용하면 예제 2.1과 같은 경우 (그림 3)과 같은 결과를 얻을 수 있다. (그림 3)은 예제 2.1에서 유전자 팀과 같이 의미 있는 영역을 표시한 것이다.

일반적으로 두 유전체 사이의 유사 유전자들을 나타내는 맵은 각 유전체의 전체 길이가 아주 길며, 두 유전체 사이의 진화 거리가 짧을수록 유사한 유전자가 많기 때문에 가시화하여 나타내면 에지 중첩이 많아 인지하기가 힘들다. E. coli K12의 경우 유전자는 4,300여개이며 유전체의 길이는 4.6백만 쌍의 염기를 가지고 있다. (그림 4)는 E. coli K12와 M. tuberculosis H37Rv 간의 유사한 영역(유전자 영역만을 고려)들을 맵으로 나타낸 것이다.

3. 계층적 클러스터링을 이용한 유전자 팀 찾기와 유사 영역의 간략화 가시화 방법론

본 연구에서 사용하는 데이터는 두 유전체의 유전자 위치 또는 Blast와 같은 프로그램으로 찾은 유사 서열의 결과이다. 두 유전체를 G_a, G_b라 하고, G_a, G_b 상의 i번째 유사 서열(혹은 유전자)을 g_a^i, g_b^i라 하자. 또한 G_a, G_b 사이의 서로 연관된 유사 서열(혹은 유전자)의 j번째 쌍을 $p_j = (g_a^m, g_b^n)$이라 하자. 입력된 두 유전체의 유사 서열을 이용하여 유전자 팀과 유사 서열 간략화를 이용한 가시화를 위해 계층적 클러스터링을 수행한다.

3.1 수정된 계층적 클러스터링 알고리즘

계층적 클러스터링은 처음에 각각의 데이터 점을 하나의 클러스터로 설정한 후 이들 쌍 간의 거리를 기반으로 하여 분할, 합병해 나가는 상향식(bottom-up) 방식으로 모든 점들이 하나의 대형 클러스터에 속할 때까지 그 히스토리 정보를 유지해 나가는 방법으로, 가까운 객체끼리 군집화 시키는 방법이다. 이 알고리즘에서는 우선 모든 n개의 서로 다른 그룹이라 가정한 후에 그룹 간의 유사성(similarity)을 보고 가장 유사한 두 개의 그룹을 합병해 그룹 수를 줄여가는 과정을 전체 그룹 수가 1이 될 때까지 반복한다. 군집의 병합 또는 분리되는 과정은 2차원 도면의 Dendrogram을 사용하여 간략히 표현되며 군집화 과정에서 어떤 개체가 일단 다른 군집에 속하면 다시는 다른 군집에 속하지 못한다. 계층적 클러스터링은 트리의 root에서부터 트리의 level에 따라 클러스터 수가 정해지므로 "K개의 클러스터를 만들어라."와 같은 사용자의 요구에 맞지 않을 수 있다. 이러한 계층적 클러스터링은 생물정보학 분야에서 마이크로어레이 실험 결과 유전자들의 클러스터링에 많이 사용되고 있다. 이외에도 K-means, SOM과 같은 다양한 클러스터링 기법이 사용된다. 하지만 모든 클러스터링 방법론은 초기에 모든 데이터에 대한 거리값을 계산해야하기 때문에, $O(n^2)$의 시간이 소요된다. 따라서 본 연구에서 기존의 알고리즘을 그대로 사용할 수 없다. 이를 위해 본 연구에 맞도록 계층적 클러스터링 알고리즘을 수정하여 사용한다. 수정된 클러스터링에서 사용하는 p_i와 p_j의 거리값 d_{ij}는 다음과 같이 구한다.

$$
\begin{aligned}
&p_i = (g_a^p, g_b^q),\ p_j = (g_a^{p'}, g_b^{p'}) \\
&I_{i,j} = average(interval(g_a^p, g_a^{p'}), interval(g_b^q, g_b^{p'})) \\
&s_i = similarity\_score(p_i),\ s_j = similarity\_score(p_j) \\
&S_{ij} = average(s_i, s_j) \\
&d_{ij} = distance(p_i, d_j) = (S_{ij})^{k^1} / (I_{ij})^{k^2}
\end{aligned}
\quad (1)
$$

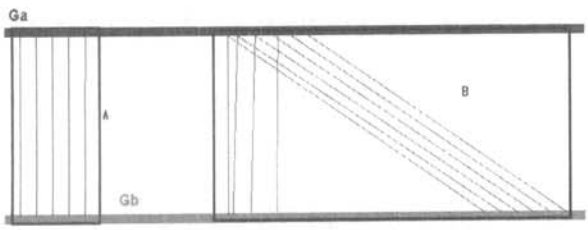
수식 1에서와 같이 유사 서열쌍들 간의 거리값은 유전체에서 유사쌍들이 가까이 있을수록 커지며 멀리 있을수록 적어진다. 수정되니 계층적 클러스터링 알고리즘의 수행은 다음과 같다.

알고리즘1 수정된 계층적 클러스터링 입력: 두 유전체 G_a, G_b 사이의 정렬 쌍의 집합 P_i 출력: 계층적 클러스터링의 결과 트리
1. 유전체 G_a의 서열 위치에 따라 정렬한다. 2. 기저 클러스터 $c_i = p_i$를 생성한다. 만약, n개의 정렬 쌍이 있다면, n개의 기저 클러스터가 생성된다. 3. 가장 가까운 거리를 갖는 쌍을 찾기 위해서 거리 값을 계산한다. 이때, p_i보다 위치상 뒤에 나타나는 것들만을 고려한다. 4. 가장 가까운 p_i, p_j를 하나의 클러스터로 만들어준 후, 거리 값을 수정하고 전체 클러스터의 수가 1이 될 때까지 3-4의 과정을 반복한다.

알고리즘 1에서 3의 가장 가까운 거리를 갖는 쌍을 찾기 위해서 거리 값을 계산을 위해서는 G_a에서의 위치에 따라 정렬하여 거리값을 계산하기 때문에, p_i와 가장 가까운 유사쌍이 p_i 이전에 나타날 경우에는 p_i를 계산하기 전에 이미 p_i가 유사쌍으로 선택되어진다. 따라서 p_i와 가장 가까운 유사쌍을 찾기 위해서는 p_i보다 뒤에 나타나는 쌍들만을 고려하면 된다. 또한 현재 p_i뒤에 나타나는 모든 쌍들을 고려하면 $O(n^2)$의 시간이 소요되므로 p_i의 이웃하는 쌍들만을 고려한다. 거리값의 계산에 필요한 k^1 파라메터는 일정 이상의 정렬 점수를 가지는 모든 유사 서열을 같은 점수로 보기 위해서 0으로 고정한다. 따라서 클러스터링의 수행 시간을 단축하기 위해서는 거리값 계산 시 모든 유사쌍들 간의 쌍을 고려하는 것이 아니라 각 유사쌍 p_i의 가장 가까운 유사쌍 p_j를 찾기 위해서 p_i의 이웃하는 유사쌍 $n(p_i)=(p_{i+1},p_{i+2},\cdots,p_{i+k})$만을 고려한다. 거리값의 계산에 필요한 p_i의 이웃 구간인 $(1,k)$값은 각 p_i마다 모두 달라진다. (그림 5)는 p_i와 그 이웃하는 유사 서열 쌍에 따른 이웃 구간의 차이를 보여준다.

이웃하는 구간 내에 조사해야 할 유사 서열 쌍의 수가 k개라면 클러스터링을 위한 거리값 계산 시 $O(kn)$의 시간이 소요되므로, 기존의 클러스터링 방법에 비하여 빠른 시간 내에 결과를 얻을 수 있다. 이의 증명은 다음과 같다.

[증명 3.1] p_i와 가장 가까운 유사 서열 쌍 p_j는 p_i의 이웃하는 구간 k내에 존재한다. p_j는 p_i와 p_{i+1}의 거리로 이루어진 구간 k 안에 존재한다. 만약, k구간 밖의 $p_{j'}$가 p_i와 가장 가까운 유사 서열쌍이라고 한다면, $d_{ij'} \leq d_{ij}$이다. 이는, $p_{j'}$은 p_j보다 더 멀리 떨어진 구간에 존재하는 유사 서열이라는 가정에 위배된다. 따라서 p_i의 가장 가까운 유사 서열 p_j는 구간 k내에 존재한다.



(그림 5) 이웃 유사 서열 쌍에 따른 이웃 구간의 차이 : A 구간에 존재하는 하나의 유사 서열 쌍 p_i의 이웃 구간은 B 구간에 존재하는 유사 서열 쌍 p_j의 이웃구간보다 적은 구간을 고려한다. 이는 거리값의 계산 시 유사 서열 쌍 p_i 쌍들의 거리 차에 의존하므로 p_i, p_j 바로 옆에 이웃하는 서열쌍 p_{i+1}, p_{j+1}이 거리의 기준값이 되어 고려할 구간을 정하는 것이다

3.2 간략화 가시화

일반적으로 두 유전체 사이의 유사 유전자들을 나타내는 맵은 각 유전체의 전체 길이가 아주 길며, 두 유전체 사이의 진화 거리가 짧을수록 유사한 유전자가 많기 때문에 가시화하여 나타내면 에지 중첩이 많아 인지하기가 힘들다. 본 연구에서는 계층적 클러스터링을 이용하여 입력된 유사 서열 쌍들을 클러스터링하고 가시화 기준에 따라 필터링 작업을 수행하여 사용자에게 가시화한다. 필터링 작업의 가시화 기준은 다음과 같다.

- 농도(density) : 클러스터 영역 내에 포함된 전체 유사 서열 쌍의 수와 클러스터에 포함되는 유사 서열의 수에 대한 비율. 이는 기준이 되는 유전체가 G_a, G_b인지 또는 둘 다 인지에 따라 값이 달라진다. 본 연구에서는 3 가지 모든 경우에 대하여 농도에 따라 간략화 가시화를 보여준다.

- 간격(interval) : 클러스터 내에 포함된 유사 서열 쌍들의 간격의 길이를 제한할 수 있다. 전체 유전체의 유사 서열 쌍을 간략화 하여 보고자할 경우에는 간격을 높여서 전체 유전체 서열을 포함할 수 있다.

- 클러스터내의 유사 서열 쌍의 수 $n(c_i)$: 클러스터 내에 포함되어 있는 유사 서열 쌍의 최대, 최소수를 결정할 수 있으며, 이에 따라 적은 수의 유사 서열 쌍을 가진 클러스터나 많은 수의 유사 서열 쌍을 가지는 클러스터는 제거할 수 있다.

(그림 6)은 인공적으로 만든 데이터에 대한 간략화 결과이다. (그림 6)의 (a)는 입력된 유사 서열 쌍을 표현한 것이고, (b)는 농도가 90%이상인 것을 보여준다. (c)는 60% 이상인 것을 보여주는 것으로 (b)에서 나타나지 않았던 영역이 나타남을 알 수 있다. 이는 G_a영역에서 농도가 떨어지기 때문이다. (d)는 모든 유사서열 쌍을 포함하는 클러스터를 보여준다.

(그림 6)에서 나타나는 색의 차이는 클러스터 내에 포함되어 있는 유사 서열 쌍들의 중첩에 따른 것이다. 클러스터 내의 유사 서열 쌍의 수를 n이라 하고, 유사 서열 쌍의 중첩 수를 $crossing(n)$이라 하면, 클러스터 i의 색 C_i는 아래와 같이 구한다.

$$C_i = rgb(0,255,0), \quad \text{if} \quad crossing(n)=0$$

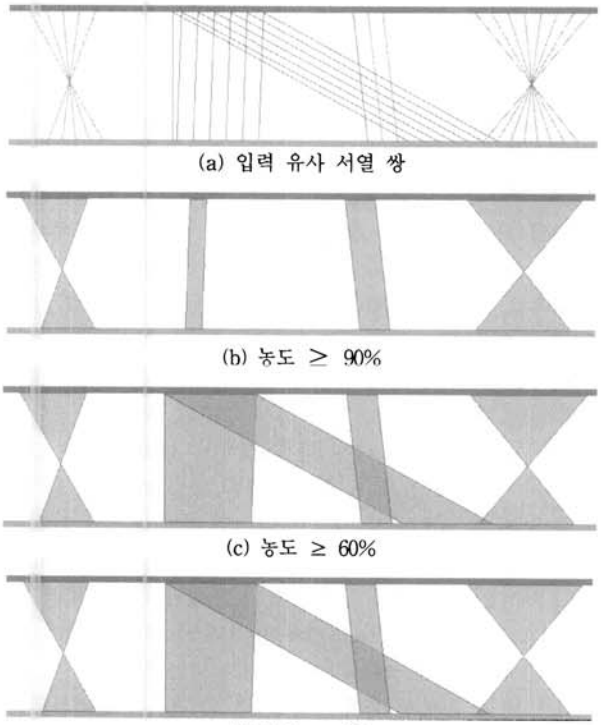
$$C_i = rbg(255,0,0), \quad \text{if} \quad crossing(n)=n(n-1)/2$$

$$C_i = rgb(255 \cdot \frac{2k}{n(n-1)}, 255 \cdot (1-\frac{2k}{n(n-1)}),0),$$

$$\text{if} \quad crossing(n)=k$$

3.3 실험 결과

본 연구에서는 E. coli K12 유전체와 M. tuberculosis H37Rv 유전체의 유전자들 간의 blast 결과를 사용하여 간



(a) 입력 유사 서열 쌍

(b) 농도 ≥ 90%

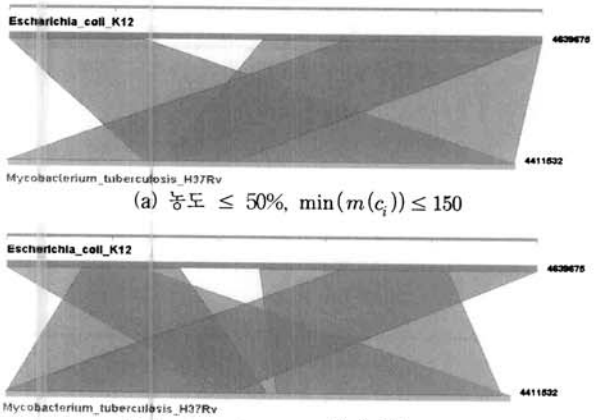
(c) 농도 ≥ 60%

(d) 농도 ≥ 0%

(그림 6) 간소화 예제 : 입력되는 유사 서열 쌍은 모두 30개이다. (a)는 입력된 유사 서열 쌍을 표현한 것이고, (b)는 농도가 90%이상인 것, (c)는 60%이상인 것, (d)는 모든 유사 서열 쌍을 포함하는 클러스터를 보여준다

략화와 유전자 팀을 찾아보았다. (그림 7)은 E. coli K12 유전체와 M. tuberculosis H37Rv 유전체의 유전자들 간의 blast 결과를 가시화한 것이다.

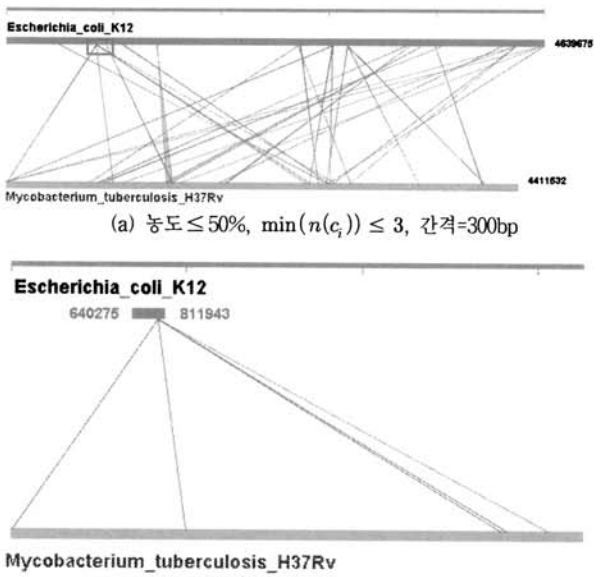
(그림 8)은 간략화 단계에서 유전자 팀을 찾아보기 위하여 유전자 팀의 일반적인 조건인 유전자 팀 내의 유전자 간의 간격을 설정하여 간략화한 것이다. (a)는 유전자 팀 내의 유전자



(a) 농도 ≤ 50%, $\min(m(c_i)) \le 150$

(b) 농도 ≤ 30%, $\min(n(c_i)) \le 150$

(그림 7) 간략화 결과 : E. coli K12 유전체와 M. tuberculosis H37Rv 유전체의 유전자들 간의 blast결과를 간략화 한 것이다. 그림 5의 가시화를 보는 것보다 간략화 하여 보는 것이 전체 유전체 간의 유사도 패턴을 인지하는 데 도움이 된다



(a) 농도≤50%, $\min(n(c_i)) \le 3$, 간격=300bp

(b) (a)의 파란 사각 영역의 확대

(그림 8) 후보 유전자 팀 가시화 : E. coli K12 유전체의 특정 영역에서 M. tuberculosis H37Rv 유전체의 여러 영역에 유사성이 높게 나타남을 볼 수 있다

간의 간격을 300bp로 설정한 결과이고, (b)는 (a)의 파란 사각형 부분을 확대한 것이다. (b)에서 나타난 후보 유전자 팀들은 E. coli K12 유전체의 특정 영역에서 M. tuberculosis H37Rv 유전체의 여러 영역에 유사성이 높게 나타남을 볼 수 있다. 이는 기존의 방법론들에 의해서는 밝혀내지 못하는 클러스터들이다.

4. 정렬 복잡도(Alignment Complexity)

두 유전체 간의 정렬은 두 유전체간의 단백질 서열이나 핵산 서열 사이의 상관관계를 나타내는 것이다. 따라서 관심 대상인 하나의 서열과 유사성(상동성)이 높은 서열들을 알아내어 그 서열의 기능을 유추하거나, 관련 있는 서열들 간의 진화적 연관성 같은 것들을 예측하기 위해서 사용된다. 물론 특별한 유사성이 없음에도 불구하고 그 구조나 기능이 유사한 경우도 종종 있다. 유전체 정렬의 결과 유사한 부분의 서열 리스트를 얻게 되는데, 각 유사한 서열간의 유사한 정도를 나타내는 점수와 통계학적 유사 점수인 p-value를 함께 제공한다. 이를 이용하여 연구자들은 각 서열들이 얼마나 유사한지를 예측하게 된다. 많은 연구자들이 두 서열간의 유사정도를 얼마나 더 잘 나타낼 수 있도록 연구하고 있다.

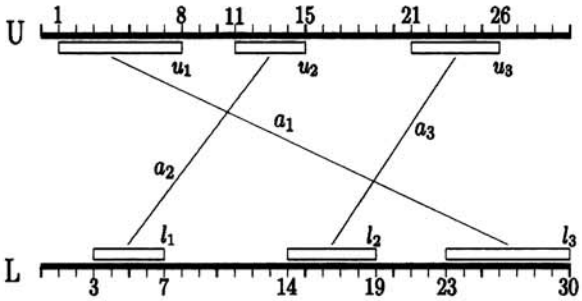
비록 정렬 결과인 유사 서열 각각의 유사정도에 따른 측정값들에 대한 연구들이 진행되고 있지만, 유사 결과 쌍들의 집합에 대한 유사도나 그 밖의 다른 분석을 위한 측정 방법은 현재까지 없는 실정이다. 본 논문에서는 유사 서열 쌍 하나하나의 유사도 정도 값을 측정하기 보다는 일정 임계값 이상의 유사 쌍들의 집합을 하나의 측정값으로 표현하는"정렬 복잡도(alignment complexity)"를 제안한다. 정렬

복잡도에 대한 설명을 하기에 앞서, 전유전체 정렬 결과 데이터들을 *이분 그래프(bipartite graph)*로 표현한다. 이는 *정렬 복잡도*를 계산할 때 필요한 몇몇 개념들이 그래프 자료구조에 기반을 두기 때문이다.

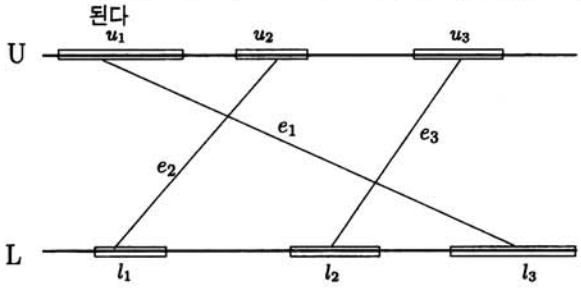
(그림 9)는 두 유전체 U와 L 사이의 정렬 쌍의 구조를 보여준다. 각 정렬 쌍은 두 유전체 사이의 서로 정렬되는 두 서열로 이루어진다. 이를 바탕으로 각 정렬 쌍 a_i를 $a_i=(u_p,l_q)$로 표현한다. u_p와 l_q는 두 유전체 사이에서 서로 정렬되는 두 서열을 뜻한다. u_p는 유전체 U에서의 p번째 위치에 있는 부분 서열이고, l_q는 유전체 L에서의 q번째 위치에 있는 부분 서열이다.

(그림 9)에서 보이는 것과 같은 유전체 정렬 결과를 AlignScope[5, 6]에서는 "순서화된 이분 그래프(ordered bipartite graph)"자료 구조로 표현한다. 순서화된 이분 그래프는 이분 그래프의 두 노드 집합 U와 L의 각각의 집합에서 노드들 간에 선형적인 순서가 존재하는 그래프 $G=(U,L,E)$를 말한다.

(그림 10)은 (그림 9)의 유전체 정렬 결과를 순서화된 이분 그래프로 표현한 것이다. 두 유전체 U와 L에서의 각 부분 서열들이 각 순서화된 이분 그래프에서의 노드가 되고 정렬 쌍이 각 에지가 된다.



(그림 9) 두 유전체 U와 L 사이의 정렬 쌍의 구조 : 유전체 U의 부분 서열 u_1은 유전체 L의 부분 서열 l_3에 정렬된다



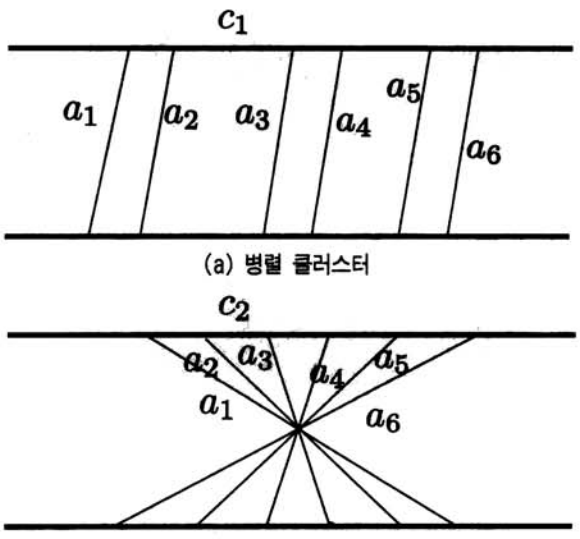
(그림 10) 순서화된 이분 그래프의 예 : (그림 9)의 각 정렬 쌍 a_i가 순서화된 이분 그래프에서 에지 E로 표현되며, 유전체 U와 L에서의 각 부분 서열들이 각 순서화된 이분 그래프에서의 노드로 표현된다

4.1 전 유전체 정렬에서의 그래프 교차 수의 의미

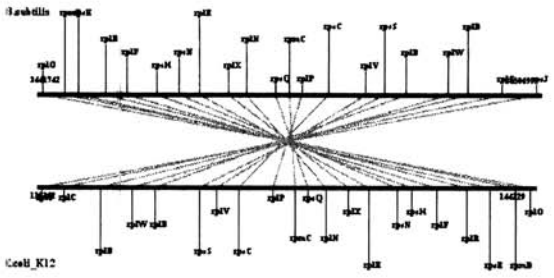
일반적으로 그래프 드로잉에서는 각 그래프의 특징을 나타내기 위한 값으로 "에지 교차 수(number of edge crossings)" 값을 사용하며, 그래프 드로잉 시 에지 교차가 많을수록 연구자가 그 그래프의 특징을 파악하기가 힘이 들기 때문에, 에지 교차수가 작은 그래프를 그렇지 않은 그래프에 비하여 좋은 그래프로 인지한다. 에지 교차수를 최소화하는 문제는 그래프 연구 분야의 고전적인 연구 문제로 많은 연구자들에 의하여 연구되어 왔다 [7, 8, 9].

(그림 11)은 두 가지 종류의 순서화된 이분 그래프에서의 에지 교차수를 나타낸다. (그림 11-(a))는 모든 에지가 평행한 경우로, 이러한 에지로 구성되는 정렬 집합을 병렬 클러스터(parallel cluster)라고 표현한다. 이와는 달리, (b)의 그래프는 6개의 에지가 존재하는 순서화된 이분 그래프에서 가장 많은 에지 교차수가 존재하는 경우로, 각 노드 집합의 연결된 노드쌍들의 순서가 각 집합에서 반대인 경우이다. 즉, U의 첫 번째 노드는 L의 마지막 노드와 연결되고, U의 두 번째 노드는 L의 마지막에서 두 번째 노드와 연결된다. 이러한 경우를 역 클러스터(reversed cluster)라고 표현하며, 이러한 경우 에지 교차수가 가장 많이 나타나게 된다. 따라서 그래프 드로잉의 관점에서 보면 (b)의 그래프에 비하여 (a)의 그래프가 더 좋은 그래프가 되는 것이다.

실제 비교 유전체 학에서는 (그림 11-(b))의 클러스터는 (a)의 그래프만큼 중요한 것이다. 이는 (b)의 두 노드 집합이 각각의 유전체를 표현한 것이므로, "U의 유전자 위치가 L에서는 완전하게 반대로 나타난다."는 뜻이 된다. 이는 U와 L 사이의 유전자들이 유전자들의 종류뿐만이 아니라 그 순서까지도 (a) 만큼이나 잘 보존되어 있다는 뜻이 된다. 서로 다른 유전체에서의 같은 유전자들이 그 순서까지 잘 보존되어 있다는 것은 진화적으로 아주 가까운 유전체임을 또는 해당 유전자 집합들이 그만큼 순서가 중요하게 보존되어 감을 예측할 수 있다. 따라서 유전체 정렬 데이터를 분석할 때에는 일반적인 그래프에서의 측정값인 에지 교차수보다 그 결과를 잘 표현할 수 있는 측정 방법이 필요하다.



(그림 11) 클러스터 내에서의 에지 교차수의 예



(그림 12) 두 유전체 *B. subtilis*와 *B. hanodurans*의 유전자 팀의 예

(그림 12)는 실제 두 유전체 *B. subtilis*와 *B. halodurans* 사이의 유전자 클러스터를 표현한 것이다[8]. (그림 12)에서 *B. subtilis*의 유전자들의 순서가 *B. halodurans*의 유전자들과 반대 방향임을 알 수 있다. 이 경우, 에지 교차 수는 최대가 되지만, 생물학적으로 이는 유전자들의 종류뿐 만이 아니라, 그 순서까지도 아주 잘 보존되어 있음을 알 수 있다 이는 단 한 번의 유전자들의 위치 이동으로 둘 사이의 유전자들의 위치가 같아 질 수 있기 때문이다.

4.2 정렬쌍 순서의 보존성을 위한 정렬 복잡도

정렬 쌍들이 잘 보존되어 있다는 의미는 하나의 유전체에 포함된 정렬 서열들의 순서가 다른 유전체의 같은 정렬 서열들의 순서와 같은 방향으로 또는 반대 방향으로 같다는 것을 뜻한다. 반면, 정렬 쌍들의 순서가 잘 보존되어 있지 않다는 의미는 두 유전체 사이의 정렬 서열들의 순서사이의 관계가 랜덤하다는 것을 의미한다.

정렬 복잡도는 각 정렬 쌍과 정렬 집합 모두 사용할 수 있으며, 정렬 집합의 정렬 복잡도 값은 그 집합에 포함된 정렬 쌍들의 정렬 복잡도의 평균을 사용한다. 그리고 두 유전체 U와 L 사이의 정렬 쌍 a_i의 정렬 복잡도, $comp(a_i)$ 값은 a_i의 양 옆에 있는 a_{i-1}과 $a+i_1$의 정렬 쌍의 순서에 의하여 결정되어진다. 세 정렬 쌍 a_{i-1}, a_i, a_{i+1}은 정렬 복잡도를 계산 할 때, 하나의 유전체 U 또는 L의 위치에 의해 정렬된 것이며, U에 의하여 정렬된 경우 a_i의 정렬 복잡도는 다음과 같이 계산된다.

$$s = 2,\ (\text{if}\ (q-p) \geq 0)\ \text{or}\ 1,\ (\text{if}\ otherwise) \quad (1)$$

$$t = 2,\ (\text{if}\ (r-q) \geq 0)\ \text{or}\ 1,\ (\text{if}\ otherwise) \quad (2)$$

$$comp(a_i) = (-1)^s \cdot 1/2^{|q-p|} + (-1)^t \cdot 1/2^{|r-q|} \quad (3)$$

수식 1과 2의 변수 s와 t는 해당 클러스터가 서로 병렬적인 관계인지 역 관계에 있는지를 알아보기 위한 값으로, 병렬적이면 2의 값을, 그렇지 않은 경우는 1의 값을 갖게 된다. 따라서 수식 3에서 s와 t의 값에 따라, $comp(a_i)$의 값의 부호가 결정된다. 양의 값을 갖게 되면 병렬적인 성향이 강한 것이고, 음의 값을 갖게 되면 역인 성향이 강한 것이다. 수식 1,2,3에서 정렬 복잡도는 3개의 정렬 쌍이 존재할 때 계산된다. 하지만 클러스터 내의 가장자리에 위치한 정렬 쌍인 경우 이웃하는 정렬 쌍이 단 하나만 존재하게 된다. 이럴 경우에는 이웃하는 정렬 쌍 하나의 값을 두 배 하여 계산한다.

수식 3에 의해서 $comp(a_i)$의 값은 -1에서 1까지의 값을 갖게 된다. 그리고 클러스터 내의 두 유전체 사이의 정렬 쌍들의 순서가 높게 보존되어 있을수록 그 절대값이 1에 가깝다. (그림 13)은 클러스터와 각 정렬 쌍의 정렬 복잡도 값의 계산을 보여주기 위한 예이다. (그림 13에는 모두 3개의 클러스터가 존재하고, 각 클러스터에는 각각 3, 3, 4개의 정렬 쌍가 존재한다.

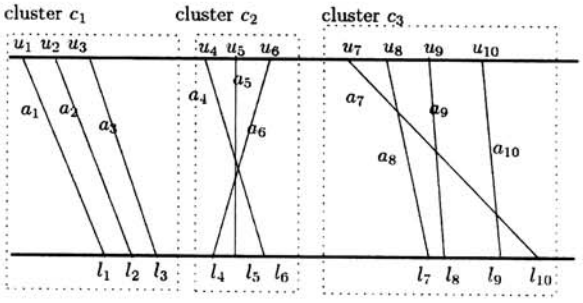
(그림 13)에서 정렬 쌍 a_2, a_6, a_9의 정렬 복잡도 값은 다음과 같이 계산된다.

- $comp(a_2) = (-1)^2 \cdot 1/2^1 + (-1)^2 \cdot 1/2^1 = 1$
- $comp(a_5) = (-1)^1 \cdot 1/2^1 + (-1)^1 \cdot 1/2^1 = -1$
- $comp(a_8) = (-1)^1 \cdot 1/2^3 + (-1)^2 \cdot 1/2^1 = 3/8$

(그림 13)의 세 클러스터의 정렬 복잡도는 다음과 같다.

- $comp(c_1) = \frac{1}{3} \cdot \sum_{i=1}^{3} comp(a_i) = 1$
- $comp(c_2) = \frac{1}{3} \cdot \sum_{i=4}^{6} comp(a_i) = -1$
- $comp(c_3) = \frac{1}{4} \cdot \sum_{i=7}^{10} comp(a_i) = 0.53125$

세 클러스터의 정렬 복잡도를 살펴보면, 클러스터 c_1과 c_2가 높은 보존성을 보이고, c_3의 보전성이 떨어지는 것으로 나타난다. 하지만 세 클러스터의 에지 교차수 만을 고려한다면, c_1은 0, c_2는 3, c_3은 2가 된다. 따라서 c_3가 가장 나쁜 클러스터가 된다. 따라서 정렬 복잡도 값이 보다 생물학적으로 의미가 있음을 알 수 있다.



(그림 13) 클러스터와 정렬 쌍의 정렬 복잡도 계산을 위한 예

4.3 실험

본 문서에서는 두 유전체 사이에서의 정렬 복잡도의 의미

를 파악하기 위해서 두 가지 경우로 나누어서 테스트하였다.

(1) 두 유전체의 유전자 팀 내에서의 정렬 복잡도
(2) 두 유전체의 정렬 복잡도가 높은 구간의 유전자들의 의미

우선 유전자 팀 내에서의 정렬 복잡도를 알아보기 위해서 *E. coli K_12*와 *B. subtilis*와 *A. fulgidus*와 *M. thermautotrophicus* 내에서의 유전자 팀들을 살펴보았다.

4.3.1 E. *coli K_12*와 *B. subtilis*의 정렬

(그림 14)는 *E. coli K_12*와 *B. subtilis*의 유전자 팀 중 하나이다. (그림 14)의 클러스터에는 모두 9개의 유전자 쌍이 존재하고 모두 *memberane-bound ATP synthase*에 관련된 기능을 한다.

(그림 14)의 클러스터의 농도는 1.0이고, 정렬 복잡도 또한 1.0이다. 따라서 (그림 14)의 클러스터는 유전자들의 보전성이 아주 높으며, 그 유전자의 순서 또한 잘 보존되어 있다고 볼 수 있다. 이는 본 유전자들의 순서가 그 기능을 수행함에 있어 중요할 수도 있음을 알 수 있다.

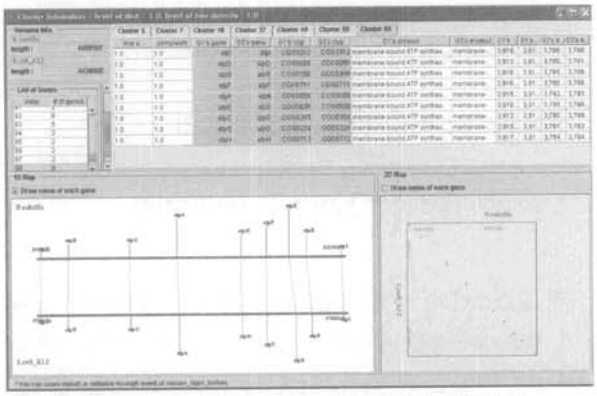
(그림 15)는 *E. coli K_12*와 *B. subtilis*의 또 다른 유전자 팀을 나타낸다. (그림 15)의 유전자들의 기능은 *flagellar biosysnthesis*에 관련된 것으로, *line density*는 0.95이고, 정렬 복잡도 또한 0.95로 그 유전자의 내용뿐만이 아니라 순서 또한 아주 잘 보존되어 있다.

4.3.2 *A. fulgidus*와 *M. thermautotrophicus*의 정렬

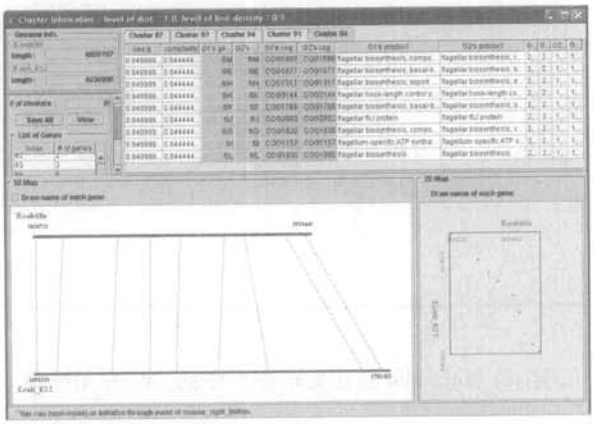
(그림 16)은 *A. fulgidus*와 *M. thermautotrophicus*의 유전자 팀을 나타낸다. (그림 16)의 클러스터는 *ribosomal protein*에 관련된 27개의 유전자로 이루어져있다. <표 1>은 (그림 16)의 클러스터 내에 포함된 유전자들을 나타낸다.

본 클러스터의 농도는 0.98이고, 정렬 복잡도 또한 -0.907이다. 따라서 (그림 16)의 클러스터는 reversed하고 그 유전자들의 순서는 거의 완전하게 반대로 되어 있다. 따라서 유전자의 내용뿐만이 아니라, 그 순서 또한 잘 보존되어 있음을 알 수 있다.

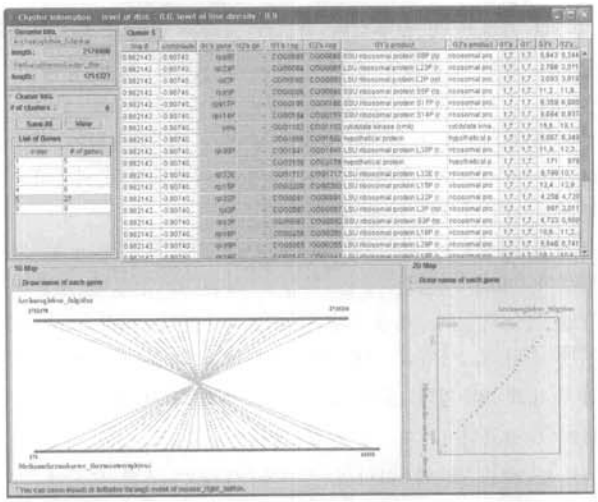
(그림 17)은 *A. fulgidus*와 *M. thermautotrophicus*의 유전자 팀을 나타낸다. (그림 17)의 클러스터는 *type 1 restriction-*



(그림 14) *E. coli K_12*와 *B. subtilis*의 유전자 팀의 예 1: *memberane-bound ATP synthase*에 관련된 기능을 하는 9개의 유전자로 이루어져 있다. (구간 [2010722~2018443],[1691529~1701411]



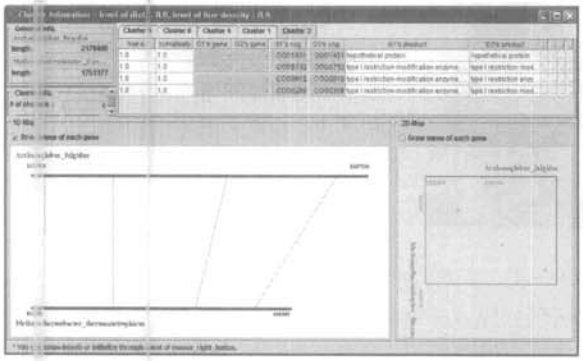
(그림 15) *E. coli K_12*와 *B. subtilis*의 유전자 팀의 예 2 : *flagellar biosysnthesis*에 관련된 기능을 하는 9개의 유전자로 이루어져 있다



(그림 16) *A. fulgidus*와 *M. thermautotrophicus*의 유전자 팀의 예 2 : *type 1 restriction-midification enzyme*에 관련된 4개의 유전자로 이루어져 있다

〈표 1〉 (그림 16)의 클러스터내에 포함된 유전자 리스트

Index	COG	*A. ulgidus*의 유전자	*M. thermautotrophicus*의 유전자	산물의 설명
1	COG0201	secY	secY	protein translocase, subunit SEC61 alpha
2	COG0096	rp23P	rp2P	LSU ribosomal protein L23P (rpl23P)
.	.	.	.	.
16	COG1588	-	-	RNAse P protein subunit P29
17	COG0092	rps3P	rps3P	SSU ribosomal protein S3P (rps3P)
18	COG0091	rpl22P	rpl22P	LSU ribosomal protein L22P (rpl22P)
19	COG0185	rps19P	rps19P	SSU ribosomal protein S19P (rps19P)
20	COG0090	rpl2P	rpl2P	LSU ribosomal protein L2P (rpl2P)
21	COG0089	rpl23P	rpl23P	LSU ribosomal protein L23P (rpl23P)
22	COG0088	rpl4P	rpl4P	LSU ribosomal protein L4P (rpl4P)
27	COG0087	rpl3P	rpl3P	LSU ribosomal protein L3P (rpl3P)



(그림 17) *A. fulgidus*와 *M. thermautotrophicus*의 유전자 팀의 예 2 : *type 1 restriction-midification enzyme*에 관련된 4개의 유전자로 이루어져 있다

*midification enzyme*으로, 4개의 유전자로 이루어져있으며, 농도와 정렬 복잡도는 모두 1.0이다. 따라서 (그림 17)의 클러스터는 유전자들의 보전성이 아주 높으며, 그 유전자의 순서 또한 잘 보존되어 있다고 볼 수 있다.

4.3.3 *A. fulgidus*와 *M. thermautotrophicus*의 정렬 복잡도

*A. fulgidus*와 *M. thermautotrophicus*의 두 유전체 사이에서 정렬 복잡도가 높은 구간을 찾아보았다. 유전자 팀처럼 클러스터가 정해져있는 부분에서의 정렬 복잡도는 계산해야할 정렬 쌍들이 정해져있지만, 이런 구간이 정해져있지 않으므로 임의의 윈도우 사이즈를 정해서 구간을 정하였다.

<표 2>는 윈도우 사이즈에 따른 0.9보다 큰 정렬 복잡도를 가지는 클러스터들을 나타낸다. <표 2>의 "*A. fulgidus* 기준에서"는 *A. fulgidus* 기준에서 0.8보다 높은 정렬 복잡도를 가지는 클러스터의 수이고, "*M. thermautotrophicus*의 기준에서"는 *M. thermautotrophicus*의 기준에서 0.9보다 높은 정렬 복잡도를 가지는 클러스터의 수를 나타낸다.

<표 2>에서 윈도우 사이즈가 9일 때는 *A. fulgidus* 기준에서 0.9보다 높은 정렬 복잡도를 가지는 클러스터의 수가 4였지만, 윈도우 사이즈가 15일 경우에는 14개로 증가한다. 이것은 유전자의 위치가 약간의 변동이 있는 클러스터가 많기 때문이다. <표 3>은 윈도우 사이즈가 10이고, 정렬 복잡도가 0.95인 클러스터의 예이다.

<표 3>의 클러스터는 (그림 16)에 나타난 *A. fulgidus*와 *M. thermautotrophicus*의 유전자 팀의 일부이다. 이를 통해,

〈표 2〉 *A. fulgidus*와 *M. thermautotrophicus*의 두 유전체 사이에서 정렬 복잡도의 수

Index	윈도우 사이즈	*A. fulgidus* 기준에서	*M. thermautotrophicus*의 기준에서
1	5	15	22
2	7	5	8
3	9	4	4
4	12	7	7
5	15	14	5

〈표 3〉 *A. fulgidus*와 *M. thermautotrophicus*의 두 유전체 사이에서 윈도우 사이즈가 10이고, 정렬 복잡도가 0.95인 클러스터의 예

Index	cog	산물의 설명
1	COG1920	hypothetical protein
2	COG0552	signal recognition particle receptor (dpa)
3	COG1730	c-myc binding protein, putative
4	COG2157	LSU ribosomal protein LXA (rplXA)
5	COG1976	hypothetical protein
6	COG2097	LSU ribosomal protein L31E (rpl31E)
7	COG2167	LSU ribosomal protein L39E (rpl39E)
8	COG2118	hypothetical protein
9	COG2238	SSU ribosomal protein S19E (rps19E)
10	COG1534	hypothetical protein

정렬 복잡도 값을 통하여, 유전자 팀들 중에서 그 순서가 잘 보존되어 있는 클러스터들을 찾을 수 있음을 알 수 있다.

5. 결 론

본 연구의 목표는 두 유전체 사이의 유사 영역의 특징을 파악하기 쉽도록 간략화 가시화를 제공하며, 간략화 단계를 통하여 기존의 방법에서 찾을 수 없었던 두 유전체 사이의 유전자 팀을 찾는 것이다. 이를 위해 계층적 클러스터링 방법론을 수정하여 사용하였으며, 몇몇 가시화 필터를 사용하여 간략화 결과를 사용자에게 제공하였다. 또한 간략화 단계를 통하여 후보 유전자 팀들을 찾아낼 수 있었다.

또한 본 논문에서는 두 유전체 간의 정렬 결과인 유사서열 쌍들의 집합의 보존성을 측정할 수 있는 측정값인 정렬 복잡도를 설명하고, 이를 실제 유전체를 이용하여 테스트해보았다. 본 테스트를 통해서 정렬 복잡도 값이 1에 가까운 즉, 유전자들의 순서가 잘 보존되어 있는 클러스터들을 찾아보고, 그것들의 의미를 알아보았다.

테스트 결과 몇몇 의미들을 찾아내었지만, 처음 정렬 복잡도의 테스트 시의 기대와 같이 정렬 복잡도를 이용하여 많은 양의 유전자 팀과 같은 의미 있는 부분들을 찾아내지는 못하였다. 이는 클러스터 내에 약간의 노이즈, 즉 클러스터 내에 존재하는 특정 쌍의 유전자가 어느 한쪽의 유전체에서 클러스터 내에 포함되어 있는 다른 유전자들과 실제 위치가 떨어져있는 경우 정렬 복잡도 값이 현저히 떨어지기 때문이다. 따라서 정렬 복잡도의 높은 값을 이용하면, 유전자들의 순서가 잘 보존되어 있을 뿐만 아니라, 실제 위치가 가까운 클러스터들을 찾게 된다.

추후 계층적 클러스터링 방법 외 SOM(Self-Organizing Maps), K-Mean, PCA(Principal Component Analysis) 클러스터링과 BIRCH(Balanced Iterative Reducing and Clustering using, Hierarchies)과 같은 보다 최신의 클러스터링 방법을 수정하여 본 연구에 사용함으로써 두 유전체의 정렬 쌍의 수나 특징에 맞추어 다양한 클러스터링 방법을 사용할 수 있도록 시스템을 확장할 것이다.

6. 감사의 글

이 논문은 2006년도 정부재원(교육인적자원부 학술연구조성사업비)으로 한국학술진흥재단의 지원을 받아 연구되었음(KRF-2006-521-D00379). 그리고 연구자료 수집과 정리에 도움을 준 정우근 연구원에게 본 논문 저자들은 각별한 감사의 말을 전합니다.

참 고 문 헌

[1] R. Overbeek, M. Fonstein, M. D'Souza, G. D. Pusch, and N. Maltsev, The use of gene clusters to infer functional cupling, In Proc, the National Academy of Sciences USA, 1999.

[2] S. Goto H. Ogata, W. Fujibunchi and M. Kanehisa, A heuristic graph comparison algorithm and its application to detect functionally related enzyme clusters, NAR, 2000.

[3] Tatusov RL, Fedorova ND, Jackson JD, Jacobs AR, Kiryutin B, Koonin EV, Krylov DM, Mazumder R, Mekhedov SL, Nikolskaya AN, Rao BS, Smirnov S, Sverdlov AV, Vasudevan S, Wolf YI, Yin JJ, and Natale DA, The cog database: an updated version includes eukaryotes, BMC Bioinformatics, 2003.

[4] Risler J.-L. Bergeron A. Nicolas, L and M. Raffinot, Gene teams: a new formalization of gene clusters for comparative genomics, Computational Biology and Chemistry, 2002.

[5] Hee-Jeong Jin and Hye-Jung Kim and Jung-Hyun Choi and Hwan-Gue Cho, AlignScope : A Visual Mining Tool for Gene Team Finding with Whole Genome Alignment, 4th Asia Pacic Bioinformatics Conference, pp.69-78, 2006.

[6] Hee-Jeong and Hwan-Gue Cho, Hierarchical Alignment Graph for Gene Teams Finding on Whole Genomes, SAC(the acm Symposium on Applied Computing), pp.113-117, 2007.

[7] M. Junger and P. Mutzel, 2-layer straightline crossing minimization: performance of exact and heuristic algorithms, Journal of Graph Algorithms and Applications, pp.1-25, 1997.

[8] P. Eades and N. Wormald, Edge crossings in drawings of bipartite graphs, Algorithmica, pp.379-403, 1994.

[9] A. Yamaguchi and Sugimoto, An approximation algorithm for the two-layered graph drawing problem, Proc, of the 6th Annual International Computing and Combinatorics Conference, Lecture Notes in Computer Science, pp.81-91, 1999.

진 희 정

e-mail : hjjin@kiom.re.kr

2000년 부산대학교 전자계산학과(학사)

2002년 부산대학교 전자계산학과(석사)

2002년~2003년 국립보건원 유전체센터 생물정보학팀

2006년 부산대학교 컴퓨터공학과(박사)

2007년~현 재 한국한의학연구원 선임연구원

관심분야 : 생물정보학(비교유전체학, 단백질 상호작용 데이터 분석)

박 수 현

e-mail : shpark@pusan.ac.kr

2007년 부산대학교 전자전기정보컴퓨터공학부(학사)

2009년 부산대학교 컴퓨터공학과(석사)

2009년~현 재 삼성전자

관심분야 : 3D 그래픽스, 응용 그래프 이론

조 환 규

e-mail : hgcho@pusan.ac.kr

1984년 서울대학교 계산통계학과(석사)

1986년 KAIST 대학원 전산학과(공학석사)

1990년 KAIST 대학원 전산학과(공학박사)

1990년~현 재 부산대학교 정보컴퓨터공학부 교수, 한국정보올림피아드 운영위원

관심분야 : 그래픽스, 알고리즘 설계와 분석, 응용 그래프 이론, 생물정보학