서픽스트리 클러스터링 방법과 블라스트를 통합한 유전자 서열의 클러스터링과 기능검색에 관한 연구

A Study on Clustering and Identifying Gene Sequences using Suffix Tree Clustering Method and BLAST

한 상 일, 이 성 근, 김 경 훈, 이 주 영, 김 영 한, 황 규 석*
(Sang il Han, Sung Gun Lee, Kyung-Hoon Kim, Ju Yeong Lee, Young Han Kim, and Kyu Suk Hwang)

Abstract : The DNA and protein data of diverse species have been daily discovered and deposited in the public archives according to each established format. Database systems in the public archives provide not only an easy-to-use, flexible interface to the public, but also in silico analysis tools of unidentified sequence data. Of such in silico analysis tools, multiple sequence alignment [1] methods relying on pairwise alignment and Smith-Waterman algorithm [2] enable us to identify unknown DNA, protein sequences or phylogenetic relation among several species. However, in the existing multiple alignment method as the number of sequences increases, the runtime increases exponentially. In order to remedy this problem, we adopted a parallel processing suffix tree algorithm that is able to search for common subsequences at one time without pairwise alignment. Also, the cross-matching subsequences triggering inexact-matching among the searched common subsequences might be produced. So, the cross-matching masking process was suggested in this paper. To identify the function of the clusters generated by suffix tree clustering, BLAST was combined with a clustering tool. Our clustering and annotating tool is summarized as the following steps: (1) construction of suffix tree; (2) masking of cross-matching pairs; (3) clustering of gene sequences and (4) annotating gene clusters by BLAST search. The system was successfully evaluated with 22 gene sequences in the pyrubate pathway of bacteria, clustering 7 clusters and finding out representative common subsequences of each cluster

Keywords : clustering, suffix tree, gene, BLAST, database

I. 서론

최근 몇 년 동안 컴퓨터를 비롯한 실험장비와 실험기술이 발달함에 따라, 여러 생물에 관련된 genomic data가 급속히 증가하였고, data를 신속하고 정확하게 분석할 수 있게 되었다[3]. 다양한 생물들의 DNA와 protein sequence data들이 전 세계에서 실험에 의해서 밝혀지고 각자의 정해진 포맷에 따라 NCBI, GenBank나 Swissprot, Unigene 같은 public database에 저장된다. 저장된 data들은 인터넷을 통해 쉽게 접근할 수 있게 구조화 되어있고, 컴퓨터를 이용한 *in silico* 방법으로 서열 data를 분석하고 의미 있는 결과를 찾는 것이 가능하다[4].

Gene sequence는 이러한 public database [5]에서 얻을 수 있는 대표적인 data로써 specific functional product (protein or RNA molecule)를 암호화하는 염색체에 존재하는 네 개의 nucleic acid (a, t, g, c)의 조합으로 된 sequence이다. 이러한 gene은 한 번에 sequencing 하기엔 너무 크기 때문에 vector를 통해 잘게 잘려진 EST 조각들을 overlap 하여서 sequencing 한다. Gene은 functional product를 암호화하는 최소단위로써, 유사한 기능을 나타내는 product를 암호화하는 gene은 진화단계에서 조상이 같을 가능성이 크기 때문에, 서로 다른 종이나 같은 종들에서 보존된 common subsequence들을 가진다[5]. 이러한 다른 서열간의 유사성을 보이는 서열들을 homologous sequence라 하고, 같은 homologous sequences에 속하는 유사한 서열들을 그룹화하는 것을 clustering 이라 한다. clustering을 통해서 기능이 밝혀지지 않은 유전자의 기능을 예측할 수 있다. 만약 다른 종끼리 유사한 유전자가 발견되면 이는 종분화에 의해서 발생한 orthologs라 하고 homologous origin과 homologous activity를 가진다. 같은 종끼리 유사한 유전자가 발견되면 이는 gene duplication에 의해 발생한 paralogs라 하고 homologous origin을 의미하지만 heterologous activities를 나타낸다.

Suffix tree 알고리즘은 선형 시간 O(n) time에 전체 data에서 matching 되는 스트링을 찾을 수 있고, 선형 공간 O(n) space에 저장될 수 있으므로, genomic data 같은 대용량 데이터를 다루는데 적절하다. Delcher [6,7]는 suffix tree를 이용하여 관련된 종들의 두 개의 genome에서 공통되는 Maximal Unique Matching subsequence(MUMs)을 찾아 배열하는 MUMmer 라는 프로그램을 만들었고, Volfovsky [8]등은 genome sequences에서 repeats를 빠르게 찾아내어서 new rice repeat database를 만들기위해 suffix tree를 적용하였다. 그들은 단지 common subsequences를 찾아내기 위해 suffix tree를 사용하였지만, 본 연구에서는 찾아낸 common subsequences를 바탕으로 유사한 서열들을 클러스터링 하는데에 suffix tree 알고리즘을 적용하였다.

Kalyanaraman [9] 등은 parallel EST clustering 프로그램을 만들었다. 서열들끼리 공유하는 maximal common substrings를 발견하였고, threshold value 이하의 길이를 보여주는 common substring을 공유하는 서열 쌍들을 pairwise alignment에 의해 비교하고 클러스터링 하였다. Zamir [10]등은 web 문서를 클러스터링하기 위해 suffix tree를 도입하였고, STC(Suffix Tree

* 책임저자(Corresponding Author)
논문접수 : 2005. 8. 25., 채택확정 : 2005. 9. 12.
한상일, 이성근, 김경훈, 이주영, 황규석 : 부산대학교 화학공학과
(sangilh@pusan.ac.kr/lee-73@pusan.ac.kr/khokim@pusan.ac.kr/juyeong@skcorp.com/kshwang@pusan.ac.kr)
김영한 : 동아대학교 화학공학과(yhkim@mail.donga.ac.kr)
※ 본 연구는 두뇌한국 21(BK21) 사업의 지원으로 연구되었음.

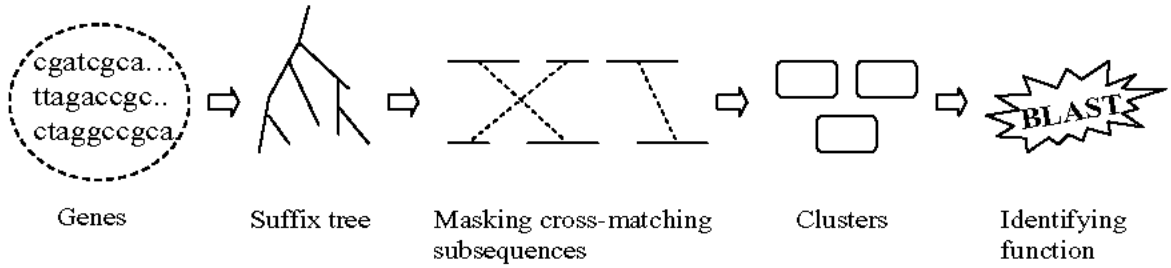

그림 1. 시스템 흐름도.
Fig. 1. System process flow.

Clustering) 방법이 기존의 클러스터링 방법에 비해서 정확성과 속도 면에서 우수하다는 것을 보였다. 본 연구에서는 Zamir가 제안한 STC 방법을 적용하여 gene sequence에 맞게 수정하였다[11]. Gene sequence를 다루는 경우에 있어서, common subsequences는 순차적으로 매치되어야 하므로, cross-matching subsequences를 제거하는 과정을 추가 하였고, 마지막으로 클러스터링된 gene sequences의 기능을 확인하기 위해서 BLAST [12] 방법을 추가하여 기존의 생물학 데이터베이스를 검색하였다.

II. 방법

Web documents를 clustering하는 STC 의 절차는 다음과 같다.

Step 1 : 문서 변환 (텍스트를 나타내는 각각의 문서의 문자가 변환된다.)

Step 2: 기준 클러스터들을 확인 (공통 문구를 공유하는 문서들의 집합들을 검색한다.)

Step 3 : 기준 클러스터들을 결합 (긴 공통부분을 가지는 기준 클러스터들을 결합한다.)

유전자를 클러스터링 하기 위해 STC를 적용하여, document string을 변환하는 불필요한 document "cleaning"단계는 수행하지 않고, step 2와 step 3를 수행하였다. 또한 매우 긴 길이를 가진 유전자들의 엇갈리는 공통부분을 없애고 common subsequences가 순차적으로 매치되도록 하기 위해 step 3-combining base clusters를 수정하였고, step 3를 two steps (grouping the common subsequence pairs and clustering the common subsequence pair groups) 으로 나누어서 유사한 DNA sequence들이 클러스터링 될 수 있도록 하였다.

Suffix tree 알고리즘을 이용해 여러 개의 서열들에서 공통으로 존재하는 subsequences를 찾아내고 위치 정보를 테이블화 하여서 sequences을 클러스터링하는 기준으로 사용한다. 다음의 그림 1은 본 연구에서 제안한 시스템의 흐름도를 나타낸다.

1. 서픽스 트리를 형성(constructing the suffix tree)

Suffix tree는 sequence의 매칭문제를 정확하고, 신속하게 해결 할 수 있는 자료구조이다. 1973년 Weiner에 의해 처음 제안되었으며, McCreight[13], Ukkonen은 같은 시간에 공간을 더 적게 차지하는 방법을 제시하였다. 특히 Ukkonen의 알고리즘은 구현과 이해가 쉽고 Weiner의 방법보다 공간을 더 적게 차지하므로, 대부분의 suffix tree 구축에 사용된다. 따라서 본 연구에서는 Ukkonen이 제안한 suffix tree 알고리즘을 도입하였다. suffix tree는 다음과 같은 특징을 가진다[14].

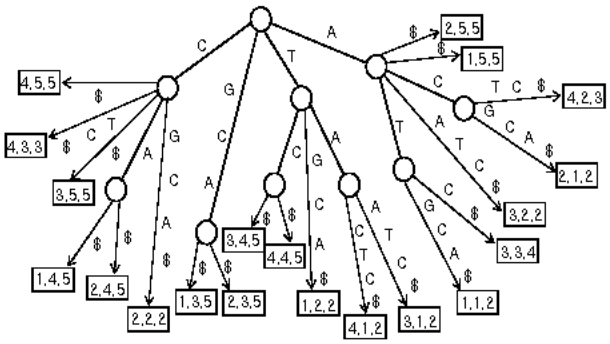

그림 2. 네개의 서열들에 대한 서픽스트리.
Fig. 2. Suffix tree for example sequences.

1) 루트와 방향성이 있는 tree 이다.

2) 길이가 m인 sequence인 경우, 1부터 m까지의 가지를 가진다.

3) 각각의 내부 node는 루트 이외에 두 개 이상의 자식 node를 가진다.

4) 가지 위에 subsequence가 label로 표시된다.

5) 같은 node에서 나오는 가지들은 똑같은 label을 가질 수 없다.

6) Nodes의 labels은 sequence의 suffix 들이다.

그림 2는 네 개의 sequences (1. ATGCA, 2. ACGCA, 3. TAATC, 4. TACTC)를 이용해 형성된 suffix tree 구조를 나타낸다. 두 개 이상의 sequence에 대한 suffix tree를 만들기 위해 각각의 sequence의 끝에 terminal symbol '$'을 첨가하여서 입력하였다. 이렇게 만들어진 suffix tree를 GST (Generalized Suffix Tree) 라 한다[15]. 그림 2의 suffix tree는 root 노드를 포함해서 모두 10개의 노드로 구성되어있고, tree 말단의 네모 상자 안의 숫자는 차례대로 각각 노드와 노드 사이의 string을 포함하는 서열의 번호와 sequence에서 string의 시작위치, 끝 위치를 나타낸다. 또한 terminal symbol $는 각각의 sequences의 끝을 나타낸다.

2. 공통 부분서열들을 검색(searching the common subsequences)

Suffix tree에서 terminal symbol '$'를 제외한, 노드와 노드 사이의 label은 두 개 이상의 서열들에서 공통으로 존재하는 common subsequence를 나타내고, 이것을 기준으로 서열의 유사성을 비교하고 클러스터링을 한다. 관계없는 불필요한 조각들로 유발된 프로그램의 비효율성을 개선하기 위해 서열

의 종류와 길이에 따라 사용자가 임의로, 시스템이 인식 가능한 최소의 subsequence길이 (minimum block size)를 선택할 수 있게 하였다.

3. 비 순차적으로 발견 되는 공통서열들을 제거(masking cross-matching subsequences)

Gene sequence의 개수가 증가하여 검색된 common subsequences들의 개수가 커지게 되면, subsequences들이 순차적으로 나열되지 않고 서로 엇갈리는 경우가 발생하여 실제 sequences들의 유사도를 저하시킬 수 있다. 따라서, 본 연구에서는 엇갈리게 매치되는 common subsequences들이 발생하였을 때, 더 긴 길이를 가지는 subsequences들을 기준으로 더 작은 길이를 가지는 subsequences들을 제거하였다. 그림 3은 이러한 엇갈리는 조각들을 제거하는 과정을 보여준다.

다음은 subsequences들이 순차적으로 매치되기 위해, cross-matching subsequences를 제거하는 규칙이다.

4. 클러스터들의 가장 긴 공통서열에 대한 blast 검색(blast search for the longest subsequences of clusters)

알려지지 않은 DNA나 단백질의 기능을 밝혀내는 것은 중요하다. 그러한 기능은 데이터베이스를 검색하여 간접적으로 밝혀질 수 있다. 그래서 본 연구에서 gene 클러스터링 시스템은 BLAST search와 결합되었다. BLAST 프로그램 (blastn, blast, blastx 등)을 사용 가능하게 하는 'blastall' tool을 NCBI (National Center for Biotechnology Information) site에서 다운받아 설치하여, GenBank, EMBL, DDBS, Swissprot databases에 대해 검색을 가능하게 하였다.(ftp://ftp.ncbi.nih.gov/blast/executables/).

Position information: two sequences(n_1 and n_2)
Pair 1{(n_1, s_1, e_1),(n_2, s_2, e_2)}
Pair 2{(n_1, s_3, e_3),(n_2, s_4, e_4)}
If the size of Pair 1 is longer, the Pair 1 is the basic pair.
Using the basic pair (Pair 1)
If ((s_1>=e_3 and s_4>=e_2) or (s_3>=e_1 and s_2>=e_4)), then
Masking cross-matching pair

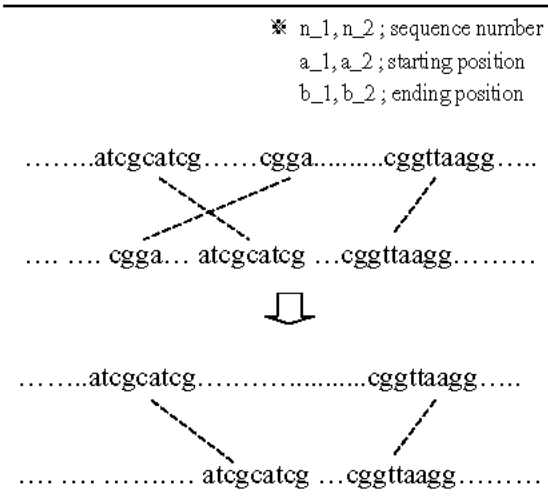

그림 3. 엇갈리는 조각들을 제거한 후, 두개의 유전자에 대한 배열.

Fig. 3. Alignments for two genes after eliminating cross-matching subsequences.

Suffix tree clustering 방법을 통해 형성된 각각의 cluster 그룹들에서 추출한 가장 긴 common subsequences들을 query data로 두고 BLAST 검색을 수행하였다. 사용자들은 필요에 의해 DNA 나 protein database에 대한 검색을 선택할 수 있다. DNA database (GenBank, EMBL, DDBJ)에 대한 검색 결과는 'nucleotide_blast.out' 파일에 저장되고, protein database에 대한 검색 결과는 'protein_blast.out' 파일에 저장된다.

III. 결과 및 토론

1. 서픽스 트리 알고리즘에 의한 유전자 서열들을 클러스터링 (clustering gene sequences by suffix tree algorithm)

본 연구에서 제안된 STC 방법은 KEGG(Kyoto Encyclopedia of Genes and Genomes) database 에서 제시된 박테리아의 pyruvate metabolic pathway와 관련된 22개의 gene sequences (표 1) 들에 대해 적용되었다. 분석은 intel's pentium 2.4 GHz processor, 512 RAM의 컴퓨터와 linux OS 환경 하에서 수행되었다. Pyruvate는 '피루브산'으로 불리고 포도당의 분해(호흡) 과정에서 생기는 물질이다. 생물체내에서 물질대사의 중간물질로 매우 중요하고, 동물이나 식물에서 탄수화물이 에너지로 이용될 때, 피루브산은 TCA cycle에 의해 산화되어 ATP 에너지를 생성하는 매개체로 작용한다.

Query gene sequences들은 fasta 파일 포맷 형태로 gene 클러스터링 프로그램에 입력된다. 프로그램을 수행하기 위해서 gene sequences들을 포함하는 파일 이름을 모니터상에 입력하고, 시스템이 인식할 수 있는 minimum block size를 입력한다. 본 연구에서는 minimum subsequence size를 10으로 설정하였다. 마지막으로 BLAST 검색을 수행할 database (DNA or protein) 형태를 선택하면 프로그램이 실행된다. 본 연구에서 시스템은 모두 7개의 cluster 그룹을 형성하였다. 22개의 gene sequences 중 *Acinetobacter_ADP1 species*와 관련된 두 개의 gene sequences를 제외하고 모두 클러스터링이 되었다. 이 두 개의 sequences들은 더 작은 크기의 common subsequences 들로 구성되므로, minimum block size 값을 작게 하면 클러스터링 될 것으로 예상된다. 클러스터링 결과 (표 2)는 KEGG database에서 ortholog 그룹과 일치하는 적절한 결과를 보여주었다. Text 파일 포맷의 파일 'cluster.out'에서 클러스터링된 gene sequences들의 결과를 볼 수 있다.

그림 4는 클러스터링된 결과의 일부분인 cluster 3을 보여준다. 사각의 점선으로 표시된 박스는 각각의 sequences에서 보존된 부분을 나타낸다. 이러한 보존된 common subsequences를 바탕으로 sequences들 간의 유사도를 비교하고 클러스터링을 하였다. cross-matching common subsequences은 부정확한 유사도를 유발할 수 있으므로, 본 연구에서는 cross-matching common subsequences들을 제거하였다. 표 3은 표 2의 각각의 cluster에서 전체 common subsequences중, 제거된 cross-matching subsequences들의 비율을 보여주었는데, cluster 3에서는 많은 서열들이 cross matching 됨을 알 수가 있다. cluster 3과 관련된 유전자가 다른 종들 간에 좀 더 복잡하게 관련되어 있다고 판단된다.

2. Blast를 이용하여 유전자 서열 클러스터들의 기능검색 (identifying gene sequence clusters by blast)

본 연구에서 gene clusters들의 기능을 파악하기 위해, 표 2

표 1. Pyruvate 대사경로의 22개 유전자.
Table 1. The 22 Genes of Pyruvate metabolic pathway.

Name	Entry
E.coli_J	JW3928, JW3366, JW1475, JW2447, JW2198, JW0111, JW0110
S.flexneri	SF4033
S.typhi	STY4296, STY1494, STY2709, TY0176, STY0175
E.carotovora	ECA0187, ECA3082
P.luminescens	plu0100, plu1546, plu3622, plu3623
C.violaceum	CV0916
Acinetobacter_ADP1	ACIAD1007, ACIAD3507

표 2. 표 1의 22개 유전자에 대한 클러스터.
Table 2. The gene clusters for the 22 genes of table 1.

Cluster	Entry
Cluster 1	JW3928, SF4033, ECA0187
Cluster 2	JW2447, STY2709, CV0916
Cluster 3	JW0111, TY0176, plu3622
Cluster 4	STY0175, JW0110, plu3623
Cluster 5	JW3366, STY4296, plu0100
Cluster 6	JW1475, STY1494, plu1546
Cluster 7	JW2198, ECA3082

의 각각의 클러스터 그룹에서 가장 긴 common subsequences 들을 추출하여 BLAST 검색을 위한 query sequences로 사용하였다(표 4). protein database를 검색하기 위해, 추출한 DNA sequences들을 protein sequences들로 번역하고 BLAST 검색을 수행하였다.

인터넷을 경유한 원격 BLAST는 불안정하고 느리기 때문에, 본 연구에서는 컴퓨터에 직접 local BLAST tool을 설치하고 사용하였다. BLAST 수행의 결과는 파일 'protein_blast.out'

표 3. 표 2의 클러스터에 대한 엇갈리는 조각 비율.
Table 3. The ratio of the cross-matching subsequences in the clusters of table 2.

Cluster	cross-matching subsequences	total subsequences	The ratio of masking (%)
Cluster 1	36	157	22.93
Cluster 2	26	143	18.18
Cluster 3	95	181	52.49
Cluster 4	19	197	9.64
Cluster 5	10	95	10.53
Cluster 6	13	76	17.11
Cluster 7	5	23	21.74

표 4. 클러스터에서 가장 긴 단백질 공통 서열.
Table 4. The longest protein subsequences in the clusters.

Cluster	Common subsequence
1	MNEQYSALRSNVSMLGKVLGETIKDALGEHIL ERVETIRKLSKSSRAGNDANRQELLTTLQNLSN DELLPVARAFSQFLNLANTAEQYHSISPKGEAA SNPEVIARTLRKLKNQPELSEDTIKKAVESLSLE LVLTAHPTEITRRTLIHKMVEVNACLKQLDNKD IADYE
2	NPEPEILPPLAKEVRP
3	EQSLITVEGDKASME
4	LNGEGLQHEDGHSHIQSLTIPNCISYDPAYA
5	IGGTWYGGEM
6	EEIIREMHK
7	FDLVKYL

에서 text 형태로 저장된다(그림 5). 그림 5에서 query sequence 'EQSLITVEGDKASME'는 protein databases 들에 대해, 모두 5개의 유사한 sequences 들이 발견되었고, 5개 서열들의 기능은' Dihydrolipoyllysine-residueacetyltransfera…'임을 알 수 있다.

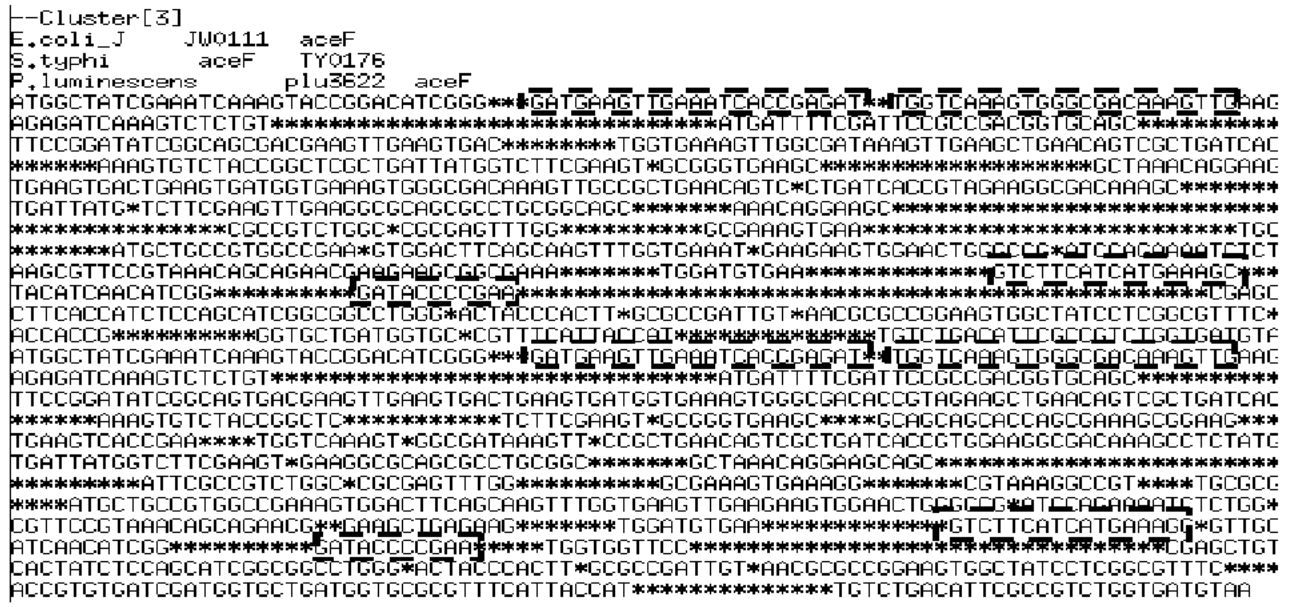

그림 4. 파일 'cluster.out'의 일부분.
Fig. 4. A part of file 'cluster.out'.

```
Query= cluster 3 EQSLITVEGDKASME
       (15 letters)

Database: /home/blast/db/swissprot
          170,940 sequences; 62,898,798 total letters

Searching.....................................................done

                                                                  Score    E
Sequences producing significant alignments:                       (bits) Value

sp|P45118|ODP2_HAEIN Dihydrolipoyllysine-residue acetyltransfera...    32   0.36
sp|P06959|ODP2_ECOLI Dihydrolipoyllysine-residue acetyltransfera...    32   0.36
sp|Q8K9T8|ODP2_BUCAP Dihydrolipoyllysine-residue acetyltransfera...    29   2.3
sp|P57302|ODP2_BUCAI Dihydrolipoyllysine-residue acetyltransfera...    29   3.1
sp|Q59638|ODP2_PSEAE Dihydrolipoyllysine-residue acetyltransfera...    27   8.9

>sp|P45118|ODP2_HAEIN Dihydrolipoyllysine-residue acetyltransferase component of pyruvate
          dehydrogenase complex (E2) (Dihydrolipoamide
          acetyltransferase component of pyruvate dehydrogenase
          complex)
         Length = 567

 Score = 32.0 bits (71), Expect = 0.36
 Identities = 15/15 (100%), Positives = 15/15 (100%)

Query: 1   EQSLITVEGDKASME 15
           EQSLITVEGDKASME
Sbjct: 137 EQSLITVEGDKASME 151
```

그림 5. 파일 'protein_blast.out'의 일부분.
Fig. 5. A part of file 'protein_blast.out'.

이것은 본래 데이터가 저장되어 있는 KEGG database에서 명명된 기능과 같으며, 따라서 본 연구의 시스템은 BLAST를 이용하여 gene clusters에 대해 기능 검색이 가능함을 보여주고 있다. cluster 6과 7을 제외하고 모두 database 검색결과에서 매치되는 sequences들을 찾아 내었다. 표 4에 나타난 가장 긴 common subsequences들은 각각의 기능을 가지는 gene clustere들을 대표하는 매우 보존된 의미있는 지역으로 예상된다. 또한, nucleotide databases들에 대한 검색 옵션을 시스템에 추가하여, protein database에 대해 유사한 sequences들을 찾지 못할 경우에, DNA수준에서도 database 검색이 가능하게 하였다. Nucleotide databases들에 대한 검색결과는 파일 'nucleotide_blast.out' 에서 제시된다.

IV. 결론

Suffix tree 알고리즘은 선형시간으로 구축이 가능한 자료구조 알고리즘으로써 대용량의 데이터를 효율적으로 다루기에 적절하다. suffix tree 알고리즘을 이용한 STC (Suffix Tree Clustering) 방법은 Zamir에 의해 web document를 클러스터링 하는데에 처음으로 이용되었다. 본 연구에서 gene sequences들을 빠르고 정확하게 클러스터링 하기 위해서 기존의 STC 방법을 적용하였고, 또한 BLAST 검색을 적용하여서 gene sequences 들을 identifying 하는 것도 가능하게 하였다.

본 연구에서 제안된 프로그램은 bacteria의 pyruvatemetabolic pathway의 실제 genomic 데이터에 적용하여 적절한 결과를 보여주었고, 각각의 gene들을 대표하는 공통되는 common subsequences들도 찾을 수 있었다.

현재의 방법은 gap penalty를 비롯한 세부적인 옵션을 고려하지 않았고, c언어로 만들어진 suffix tree 알고리즘을 제외한 다른부분은 Perl(Practical Extraction and Reporting Language) 언어로 만들어져 있지만, 향후 이러한 세부적인 옵션들이 추가되고 low level 언어로 프로그램화 된다면 더 좋은 성능을 보여줄 것으로 기대된다.

참고문헌

[1] C. Notredame and D. G. Higgins, "SAGA: sequence alignment by genetic algorithm," *Nucleic Acids Res.*, vol. 24, pp. 1515-1524, 1996.

[2] T. F. Smith and M. S. Waterman, "Identification of common molecular sequences," *J. Mol. Biol.*, vol. 147, pp. 195–197, 1981.

[3] J. Y. Chen and J. V. Carlis, "Genomic data modeling," *Information Systems*, vol. 28, pp. 287, 2003.

[4] D. W. Mount, "Bioinformatics: sequence and genome analysism," Cold Spring Harbor Laboratory Press, New York, pp. 3-5, 2001.

[5] J. M. Ostell, S. J. Wheelan and J. A. Kans, "The NCBI data model.," *Methods Biochem. Anal.*, vol. 43, pp. 19, 2001.

[6] A. L. Delcher, S. Kasif, R. D. Fleischmann, J. Peterson, O. White and S. L. Salzberg, "Alignment of whole genomes," *Nucleic Acids Res.*, vol. 27(11), pp. 2369-2376, 1999.

[7] A. L. Delcher, A. Phillippy, J. Carlton and S. L. Salzberg, "Fast algorithms for large-scale genome alignment and comparisonm," *Nucleic Acids Res.*, vol. 30(11), pp. 2478-2483, 2002.

[8] N. Volfovsky, B. J. Haas and S. L. Salzberg, "A clustering method for repeat analysis in DNA sequences," *Genome Biol.*, vol. 2, pp. 1-11, 2001.

[9] A. Kalyanaraman, S. Aluru and S. Kothari, "Parallel EST clustering," *HICOMB*, 185, 2002.

[10] O. Zamir, O. Etzioni, O. Madani and R. M. Karp, "Fast and intuitive clustering of Web documents," *In Proc. of the 3rd International Conference on Knowledge Discovery and Data Mining*, pp. 287-290, 1997.

[11] S. I. Han, S. G. Lee, B. K. Hou, S. H. Park, Y. H. Kim and K. S. Hwang, "A gene clustering method with masking cross-matching fragments using modified suffix tree clustering method," *Korean J. Chem. Eng.*, vol. 22(3), pp. 345, 2005.

[12] S. F. Altschul, W. Gish, W. Miller, E. Myers and D. J. Lipman, "Basic local alignment search tool," *J. Mol. Biol.*, vol. 215, pp. 403-410, 1990.

[13] E. McCreight, "A space economical suffix tree construction

algorithm," *Journal of the ACM*, vol. 23, pp. 262-272, 1976.
[14] E. Ukkonen, "On-line construction of suffix trees," *Algorithmica*, vol. 14, pp. 249-260, 1995.
[15] D. Gusfield, "Algorithms on strings, trees, and sequences: computer science and computational biology," Cambridge University Press, London, pp. 116, 1997.

한 상 일

1978년 4월 4일생. 2003년 부산대학교 화학공학과(학사). 2005년~현재 부산대학교 화학공학과(석사). 관심분야는 생물정보학, 시스템 미생물학.

김 경 훈

1968년 3월 1일생. 1992년 부산대학교 화학공학과(학사). 1994년 부산대학교 화학공학과(석사). 2000년 부산대학교 화학공학과(박사). 2002년 일본 JAIST post-doc. 2003년~현재 부산대학교 화학공학과 BK21 핵심분야 사업팀 post-doc.

김 영 한

1952년 8월 21일생. 1976년 동아대학교 화학공학과 학사. 1980년 한국과학기술원 화학공학과 석사. 1984년 미국 Lamar Univ. 화학공학과 박사. 현재 동아대학교 화학공학과 교수. 관심분야는 화학공정설계, 화학공정제어, 화학공정센서.

이 성 근

1973년 3월 1일생. 1996년 부경대학교 화학공학과(학사). 1998년 부산대학교 화학공학과(석사). 2005년~현재 부산대학교 화학공학과(박사). 관심분야는 생물정보학, 시스템 미생물학.

이 주 영

1969년 3월 4일생. 1993년 부산대학교 화학공학과(학사). 1995년 부산대학교 화학공학과(석사). 1999년~현재 부산대학교 화학공학과 박사과정. 관심분야는 화학공정설계, 화학공정제어.

황 규 석

1955년 1월 4일생. 1982년 부산대학교 화학공학과(학사). 1985년 일본 동경공업대학 화학공학과(석사). 1988년 일본 동경공업대학 화학공학과(박사). 현재 부산대학교 화학공학과 교수. 관심분야는 화학공장자동화, 화학공정전문가시스템, 화학공정안전, 생물정보학.