# 전유전체(Whole genome) 서열 분석과 가시화를 위한 워크벤치 개발

최 정 현[†]·진 희 정[††]·김 철 민[†††]·장 철 훈[†††]·조 환 규[††††]


## 요 약

최근 활발한 소단위 게놈 프로젝트의 수행으로 많은 생물체의 유전체 전체 서열이 밝혀짐에 따라서 전유전체(whole genome)를 기본 단위로 하여 개별 유전자나 그에 관련된 기능 연구가 매우 활발히 이루어지고 있다. 전유전체의 염기 서열은 수백만 bp(base pairs)에서 수백억 bp(base pairs) 정도의 대용량 텍스트 데이터이기 때문에 단순한 온라인 문자 일치(on-line string matching) 알고리즘으로 분석하는 것은 매우 비효율적이다. 본 논문에서는 대용량의 유전체 서열을 분석하는데 적합한 자료 구조인 스트링 B-트리를 사용하여 유전체 서열의 분석과 가시화를 위한 워크벤치를 개발한 과정을 소개한다. 본 연구에서 개발한 시스템은 크게 질의문 부분과 가시화 부분으로 나뉘어 진다. 질의문 부분에는 유전체 서열에 특정 서열이 나타나는 부분의 위치와 횟수를 알아보거나 $k$번 나타나는 서열을 조사하는 것과 같은 기본적인 패턴 검색 부분과 $k$-mer 분석을 위한 질의어가 다양하게 준비되어 있다. 가시화 부분은 전유전체 서열과 주석(annotation)을 보여주거나, 유전체 분석을 용이하도록 여러 가시화 방법, CGR(Chaos Game Representation), $k$-mer graph, RWP(Random Walk Plot) 등으로 생물학자들이 쉽게 전체 구조와 특성 파악할 수 있도록 도와준다. 본 논문이 제안하는 분석 시스템은 생물체의 진화적 관계를 밝히고, 염색체 내에 아직 알려지지 않은 새로운 유전자나 기능이 밝혀지지 않은 junk DNA들의 기능 등을 연구하는데 사용할 수 있다.


# Development of Workbench for Analysis and Visualization of Whole Genome Sequence

Jeong-Hyeon Choi[†] · Hee-Jeong Jin[††] · Cheol-Min Kim[†††]
Chul-Hun L. Chang[†††] · Hwan-Gue Cho[††††]


## ABSTRACT

As whole genome sequences of many organisms have been revealed by small-scale genome projects, the intensive research on individual genes and their functions has been performed. However on-memory algorithms are inefficient to analysis of whole genome sequences, since the size of individual whole genome is from several million base pairs to hundreds billion base pairs. In order to effectively manipulate the huge sequence data, it is necessary to use the indexed data structure for external memory. In this paper, we introduce a workbench system for analysis and visualization of whole genome sequence using string B-tree that is suitable for analysis of huge data. This system consists of two parts : analysis query part and visualization part. Query system supports various transactions such as sequence search, $k$-occurrence, and $k$-mer analysis. Visualization system helps biological scientist to easily understand whole structure and specificity by many kinds of visualization such as whole genome sequence, annotation, CGR (Chaos Game Representation), $k$-mer, and RWP (Random Walk Plot). One can find the relations among organisms, predict the genes in a genome, and research on the function of junk DNA using our workbench.


**키워드 : 문자열 일치(string matching), 스트링 B-트리(string B-tree), 서열 분석(sequence analysis), $k$-mer 분석($k$-mer anlysis), chaos game representation, $k$-mer graph, random walk plot**

## 1. 서 론

최근에 많은 게놈 프로젝트의 결과로 생물체의 서열이 밝혀짐에 따라서 전유전체(whole genome)를 기본 단위로 하여 개별 유전자나 그에 관련된 기능 연구를 위한 유전체 분석이 매우 활발히 이루어지고 있다[3, 7, 17, 21]. 유전체 분석이란 생물체가 가지는 유전정보, 즉 유전체의 염기서열의 정보를 해석하고 비교 유전체학, 단백질 분석학, 생물정보학 등 여러 학문들과 함께 유전체 상의 유전자의 위치나 그 기능 등에 대해 연구하는 분야이다. 이러한 유전체 분석을 통해 생물의 전체 유전자를 파악하게 되면 이를 바탕으로 각

※ 본 연구는 한국과학재단 목적기초연구 사업 (2000-1-30300-012-2)과 부산대학교 의학연구소 연구비 (2001-1-25)의 지원으로 수행되었음.
† 준 회 원 : 부산대학교 대학원 전자계산학과
†† 정 회 원 : 국립보건원 유전체연구소 역학정보실 생물정보학팀 선임연구원
††† 정 회 원 : 부산대학교 의학과 교수
†††† 정 회 원 : 부산대학교 전기전자정보컴퓨터공학부 교수, 컴퓨터 및 정보통신 연구소
논문접수 : 2002년 3월 5일, 심사완료 : 2002년 7월 5일

유전자의 작용을 알아내 결함을 수정하고 기능을 강화하거나 특정 기능을 다른 생물체에 적용시키는 등의 다양한 생물 공학적 응용이 가능해진다. 현재까지 유전체 분석에 필요한 많은 기능들을 제공하는 프로그램들은 대부분 상용 프로그램이다. 상용 프로그램들 중에서 연구자들이 많이 사용하고 있는 유전체 분석 툴에는 Vector NTI[23], GenoMax[11], DNASpace[9] 등이 있다.

Vector NTI suite은 분자생물학자들이 실험을 하는 데에 가장 기본이 되는 분자 디자인과 관련된 전반적인 것에 소모되는 시간과 노력을 줄이는데 도움을 주는 프로그램으로 InforMax사에서 만든 상용 프로그램이다. 여러 모듈에서 사용되는 분자가 데이터베이스 탐색기에 저장되어 관리되므로 모듈간 이동이 간편하고, 인터넷에서 사용할 수 있는 여러 프로그램들에 쿼리를 쉽게 전송할 수 있게 만들어졌기 때문에 다양한 웹 기반 프리웨어를 Vector NTI의 모듈처럼 사용할 수 있다.

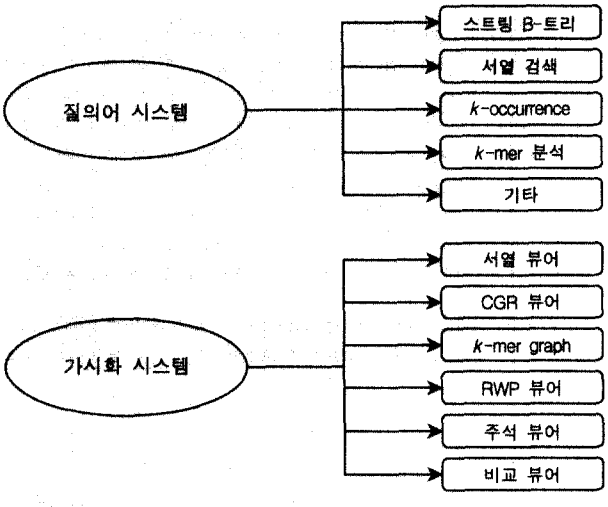
GenoMax는 오라클 데이터베이스, 서버, 자바 그래픽 사용자 인터페이스로 구성되어 있으며, VectorNTI와 마찬가지로 Informax사에서 개발된 툴이다. GenoMax에서는 서버가 모든 사용자와 데이터베이스, 데이터 마이닝 툴을 연결하고 통제한다. GenoMax는 게놈들의 통합이나 자동 탐색, 결과의 필터링, 공동 연구작업들, 확장, 분산 처리 등의 작업들을 사용할 수 있다. 따라서 GenoMax는 개인의 유전체 분석과 같은 작은 단위가 아니라, 총체적인 과정이 통합적으로 이루어져야 하는 신약 개발이나, 종 단위의 유전 정보 완성 등의 대규모 프로젝트에 주로 사용되어지고 있다.

DNASpace는 HITACHI의 MiraiBio사에서 개발된 시스템으로, 연구자의 요구에 맞춰 다양한 적용이 가능한 맞춤 바이오인포메틱스 소프트웨어로서 필요한 서열에 대한 문제를 데스크탑에서 자동적으로 해결할 수 있도록 해준다. 또한 인터넷상의 데이터베이스 검색과 함께 필요한 서열 분석과 결과 예측을 동시에 쉽게 할 수 있도록 해준다. 이 외에도 Acceleys사의 GCG Wsisonsin Package, Biomax사의 Pedant-Pro와 LION사의 SRS 등과 같은 프로그램들이 있다.

전유전체 서열을 분석하는 작업의 대부분은 유전자 영역과 같은 특정 영역이나 서열을 찾는 것이다. 이러한 작업은 전유전체 서열을 하나의 텍스트 $T$라고 하고, 우리가 찾고자 하는 유전자나 서열을 패턴 $P$라고 하면, 유전체 서열에서 특정 유전자를 찾는 문제는 텍스트 $T$에서 패턴 $P$를 찾는 것으로 볼 수 있다. 이것은 문자열 일치(string matching) 문제로 전산학에서 오랫동안 연구되어 왔다[13, 20]. 그런데, 지금까지 밝혀진 전유전체 서열은 수백만 bp(base pairs)에서 수백억 bp(base pairs)의 염기로 이루어진 대용량 텍스트이다. 이 때문에 전체 유전자에서 특정 유전자의 위치를 찾는 것과 같은 문자열 일치(string matching)를 메모리 상주(on-memory) 방식으로 처리하는 것은 비효율적이다. 따라서 대용량 데이터 처리를 위한 새로운 분석 알고리즘의 개발과 적용이 매우 필요한 상황이다.

외부 메모리에 효율적이고 염기 서열처럼 비구조적인 텍스트에 대한 색인 자료구조로 스트링 B-트리가 가장 좋은 자료구조이다. 스트링 B-트리는 외부 메모리 자료구조에 가장 좋은 B-트리와 서픽스 배열의 특성을 조합한 색인 자료구조이다[10]. 스트링 B-트리는 역파일(수정성과 atomic key), 서픽스 배열(수정성과 연속 공간), 서픽스 트리(불균형한 트리 위상), 프리픽스 B-트리(bounded-length key) 등의 제한을 모두 극복하였고, 최악의 경우에 B-트리와 같은 성능을 가지는 첫 번째 외부 메모리 자료 구조이다. 이러한 스트링 B-트리에 서픽스 배열을 이용한 빠른 생성 알고리즘과 LCP(Lonest Common Prefix)를 이용한 빠른 검색 알고리즘을 추가한 시스템을 개발하였다[5, 6]. 본 연구에서 개발된 워크벤치 시스템은 이들 연구의 내용을 유전체 분석 질의 시스템의 엔진으로 사용하여 개발되었다.

본 논문에서는 이전 연구에서 개발한 스트링 B-트리를 엔진으로 사용하여 유전체 서열의 분석과 가시화를 위한 워크벤치 시스템을 개발하였다. 이 시스템은 서열 분석을 위한 질의문 부분과 가시화 부분으로 나뉘어져 있다. 질의어 부분에는 서열 검색, $k$-occurrence, $k$-mer 분석 등이 있고, 가시화 부분에는 전유전체 서열 뷰어, 유전체 annotation 뷰어, 패턴 검색 결과 뷰어, CGR(Chaos Game Represenation) 뷰어, 온 $k$-mer 뷰어, RWP(Random Walk Plot) 뷰어 등이 있다. (그림 1)은 워크벤치의 구성을 보여준다.




(그림 1) 워크벤치의 구성도

본 논문에서 개발한 워크벤치를 이용하여 생물체별로 존재하는 특이 서열이나 존재하지 않는 서열을 조사하거나 공통 조상의 서열 등을 밝힐 수 있고, 염색체 내에 아직 알려지지 않은 새로운 유전자나 기능이 밝혀지지 않은 junk

DNA들의 기능 등 다양한 분석을 할 수 있다. 특히 각 생물 종들에게 있어 종 특이적인 패턴을 나타내는 짧은 길이의 oligonucleotide들의 빈도수, 척추동물에서의 CG dinucleotides의 결여, cyanobacteria에서의 highly iterated palindrome 1(HIP 1)의 과발현, archaebacteria인 *Archeglobus fulgidus* 유전체에서 CTAG 패턴의 결여 등을 분석할 수 있다[4, 16, 21, 22]. 이러한 염기들의 빈도 패턴은 다른 종보다 같은 종 사이에서 더 전형적인 형태를 유지하는 것으로 보고되어 있다. 이러한 유전체 서열 내에서 oligonucleotide의 패턴에 대한 연구는 인간 개개인적인 구별이 가능한 short tandem repeat(STR or microsatellite) 서열을 이용한 유전체 분석과 유전자 발현과정 중 전사조절인자의 공통된 결합부위 패턴 등이 분석된 바 있다[19].

본 논문의 구성은 2절에서 질의어 시스템의 알고리즘과 구현에 대해 설명하고, 3절에서 가시화 시스템의 기능과 알고리즘에 대해 설명하고, 4절에서 워크벤치를 이용하여 유전체 서열에서 잘 나타나지 않는 서열(avoided sequence)을 찾는 문제를 해결하고 그 결과를 보여준다.

## 2 질의어 시스템

질의어 시스템은 전유전체 서열을 분석하기 위한 여러 질의어들을 제공해주는 부분이다. 제공하는 질의어로는 유전체에서 특정 서열을 검색하는 질의어, $k$번 나타나는 서열을 검색하는 질의어, $k$-mer 분석을 위한 질의어로 나뉘어져 있다.

### 2.1 서열 검색

유전체 서열 분석의 가장 기본적인 작업이 전유전체 서열에서 특정 서열을 찾는 것이다. 서열 검색은 유전체에서 특정 서열이 존재하는지, 존재한다면 그 서열이 몇 번 나타나는가와 같은 질의를 말한다. 서열 검색을 하기 위해서는 스트링 B-트리의 루트 노트부터 검색을 시작하여 단말 노드까지 검색하여 단말노드에 있는 서픽스들에 대한 서열의 위치를 구한 다음 단말 노드를 따라가면서 일치하는 개수를 세면 된다[5]. 이것을 정리하면 (알고리즘 1)과 같다. 서열 검색 결과로 알 수 있는 것은 서열의 존재 유무(existence), 서열이 나타나는 횟수(occurrence), 서열이 나타나는 위치(position) 등이다.

| search(sbt, s) : 스트링 B-트리를 이용한 서열 검색 알고리즘 | |
|---|---|
| 입력 : | sbt- 스트링 B-트리<br>s- 서열 |
| 출력 : | count- 서열이 나타나는 횟수<br>pos- 서열이 나타나는 위치 배열 |

```
count = 0 ;
(node, j) = SB-Search-Up-Down ( s, sbt.root, 0 ) ;
while node is not NULL
  for i from j to N_leaf - 1
    s = 노드의 i번째 서픽스 ;
    lcp = LCP ( pat, s ) ;
    if lcp < | s | then
      return ( count, pos ) ;
    end if
    count = count + 1 ;
    pos에 s 추가 ;
  end for
  node = next ( node ) ;
  j = 0 ;
  node를 메모리에 로드한다. ;
end while
```

(알고리즘 1) 서열 검색 알고리즘

여기서 SB-Search-Up-Down( )은 루트 노드부터 단말 노드까지 순회하면서 서열이 단말 노드에 나타나는 위치를 구해주는 함수이고, LCP( )는 두 문자열의 LCP(Longest Common Prefix)의 길이를 구해주는 함수이고, next()는 다음 단말 노드의 페이지 번호를 구해주는 함수이다. 길이가 $n$인 서열 검색 알고리즘의 시간 복잡도는 $O(n\log_B m)$이다. 여기에서 $m$은 전체 서열의 길이이고, $B$는 디스크 페이지의 크기이다.

### 2.2 $k$-occurrence

$k$-occurrence 문제는 텍스트에서 $k$번 나타나는 패턴을 찾는 문제이다. 이 문제를 효율적으로 해결하는 방법은 스트링 B-트리의 단말 노드들이 서픽스 배열과 같다는 사실과 sliding window를 이용한다. 즉 $k$-1개의 크기를 가지는 윈

| kocc ( s, pos, lcp, k ) : 스트링 B-트리를 이용한 $k$번 나타나는 서열 검색 알고리즘 | |
|---|---|
| 입력 : | s- 서열<br>pos- 서픽스 배열<br>lcp- 인접한 서픽스들의 LCP 배열<br>k- 횟수 |
| 출력 : | count- $k$번 나타나는 서열의 개수<br>seq- $k$번 나타나는 서열의 배열 |

```
k-- ;
let sw be slide window ;
for i = 0 to k - 1
   sw.put(i, lcp[i]) ;
end for
for i = k to | pos | - 1
   int minLcp = sw.getMin ( ) ;
   if minLcp > prevLcp and minLcp > lcp[i] then
     for j = max ( prevLcp, lcp[i] ) to minLcp
         seq에 s[pos[i-1]] ··· [pos[i-1]+j] 추가 ;
   end if
   prevLcp = sw.get ( i - k ) ;
   sw.put ( i, lcp[i] ) ;
end for
```

(알고리즘 2) $k$번 나타나는 서열 검색 알고리즘

도우를 서픽스 배열을 따라 이동하면서 윈도우의 최소 LCP의 길이가 윈도우의 경계에 있는 LCP의 길이보다 큰 곳을 찾는다. 여기서 LCP란 두 문자열의 Longest Common Prefix를 말한다. 윈도우 내의 최소 LCP의 길이를 *min*이라 하고, 윈도우의 시작 앞에 있는 LCP의 길이를 *start*라 하고, 윈도우의 끝 뒤에 있는 LCP의 길이를 *end*라 하면, 윈도우 내의 $k$번 나타나는 패턴이 있을 조건은 $min > start$이고 $min > end$이다. $k$번 나타나는 패턴은 *start*와 *end* 중 큰 값의 다음부터 윈도우 내에 있는 *min*까지의 패턴으로 이루어지는 패턴들이다. 모든 $k$-occurrence를 구하기 위해서는 서픽스 배열을 따라 윈도우를 이동하면서 위의 조건이 맞는 지를 검사해야 한다. (알고리즘 2)는 위의 알고리즘에 대한 의사 코드이다. $k$번 나타나는 서열 질의는 $k$번 나타나는 서열들과 개수를 구한다. 이 알고리즘의 시간 복잡도는 $O(m)$이다. 여기에서 $m$은 전체 서열의 길이다.

### 2.3 $k$-mer 분석

$k$-mer는 유전자의 염기 서열 내의 길이가 $k$인 연속된 염기 서열이다. $k$-mer 분석은 염기서열이 가진 $k$-mer들의 과발현(over-representation)이나 미발현(under-representation) 패턴과 같은 분포나 대칭성 등을 탐색하는 것이다. $k$-mer 분석을 위해서는 모든 $k$-mer의 빈도를 구해야 하는데 이것은 스트링 B-트리를 이용하여 쉽게 구해질 수 있다[5, 6]. 스트링 B-트리의 모든 단말 노드에는 서픽스들이 사전순으로 정렬되어 있고 인접한 서픽스들의 *lcp* 정보가 있기 때문에 단말 노드를 순회하면서 $k$-mer의 빈도를 계산하는 것이 가장 빠른 방법이다. $k$를 구하고자 하는 $k$-mer의 길이라 하고 $F$를 해당 $k$-mer의 빈도가 저장되는 배열이라 할 때 (알고리즘 3)은 빈도가 0이 아닌 모든 $k$-mer들의 빈도를 구한다. 이 알고리즘의 시간 복잡도는 $O(m)$이다. 여기에서 $m$은 전체 서열의 길이다.

kmer (sbt, k) : 스트링 B-트리를 이용한 $k$-mer를 구하는 알고리즘

입력 : sbt- 스트링 B-트리
k- 횟수
출력 : freq- $k$-mer의 빈도를 저장하는 배열

```
count = 0 ;
pat = 첫 번째 k-mer ;
node = 첫 번째 단말노드 ;
while node is not NULL
    node를 메모리에 로드한다 ;
    for i from 0 to N_leaf - 1
        s = node의 i 번째 서픽스 ;
        lcp = node의 i 번째 lcp ;
        if lcp > k then
            count = count + 1 ;
        else
            if length ( s ) k then
                freq에 ( pat, count ) 추가 ;
                pat = s[0] … s[k-1] ;
            end if
            count = 1 ;
        end if
    end for
    node = next ( node ) ;
end while
```

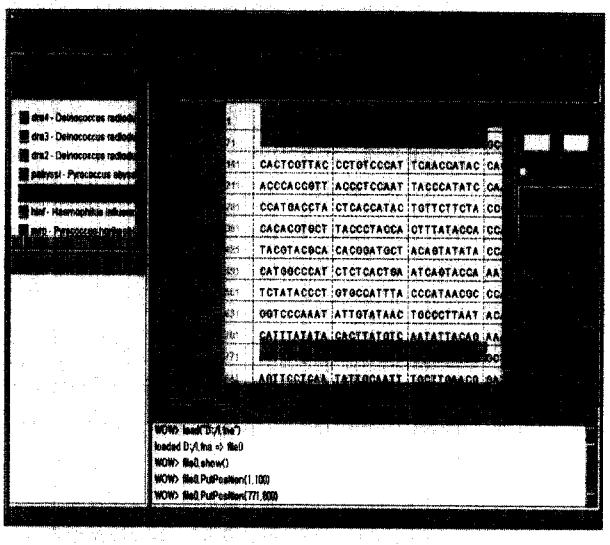
(알고리즘 3) $k$-mer 빈도를 구하는 알고리즘

## 3. 가시화 시스템

워크벤치는 유전체 분석이 용이하도록 가시화 시스템을 제공한다. 전유전체 서열은 아주 긴 서열 데이터로 그것에서 어떤 패턴을 찾거나 특정 정보를 보려고 할 때, 텍스트로 사용자에게 알려주는 것보다 가시화하여 사용자에게 보여주면 사용자가 보다 쉽게 인지할 수 있다. 가시화 시스템은 서열 뷰어, 주석 뷰어, CGR(Chaos Game Representation) 뷰어, $k$-mer 뷰어, RWP(Random Walk Plot) 뷰어, 비교 뷰어를 제공한다.

### 3.1 유전체 서열 뷰어 (Sequence Viewer)

유전체 서열 뷰어는 유전체의 서열을 텍스트로 화면에 보여주는 것이다. 이것은 유전체 분석 프로그램에서 기본적으로 제공되어야 하는 것으로, 분석하고자 하는 유전체의 염기 서열의 이중 나선구조(double strand)를 보여준다. 또한, 사용자가 특정 위치의 서열을 보고자 할 때에는 유전체 서열 뷰어에서 직접 위치를 입력하거나 명령 라인에서 위치를 입력함으로써 그 부분을 볼 수 있다. (그림 2)는 유전체 서열 뷰어의 화면이다.

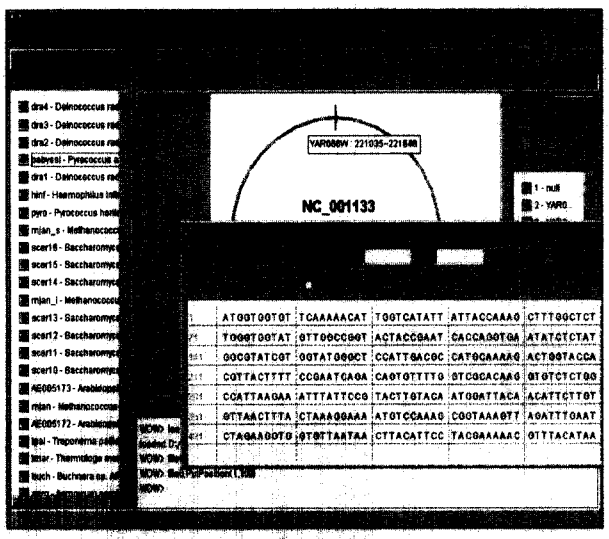


(그림 2) 유전체 서열 뷰어 : 전유전체 서열을 텍스트로 보여 주고, 이중 나선구조와 특정 서열이 나타나는 위치 정보 등을 표시할 수 있다.

### 3.2 유전체 주석 뷰어(Annotation Viewer)

유전체 서열에 대한 파일 형식은 다양한데 그 중에 Gen-

Bank 형식의 파일 있다. GenBank 형식 파일에는 유전체에 어떠한 유전자가 존재하며, 그것이 어떠한 기능을 하는 것과 같은 정보들이 저장되어 있다. 이러한 정보들을 주석(annotation)이라 한다. 주석은 유전체 분석 시 아주 유용한 것이므로 본 시스템에서는 주석을 보여주는 뷰어를 제공한다. (그림 3)은 유전체의 주석 뷰어의 화면이다.




(그림 3) 유전체 주석 뷰어 : 유전체 서열의 주석(annotation) 정보를 보여준다. 주석 뷰어에서 유전자를 클릭하면 그것의 서열을 볼 수 있다.

주석 뷰어는 (그림 3)에서 보는 것과 같이 전유전체 서열을 원형이나 선형으로 보여주고, 그 유전체에서 밝혀져 있는 유전자들의 위치를 표시한다. 유전자의 이름들은 리스트로 보여주고, 유전자 이름을 선택하면, 해당되는 유전자의 위치를 붉은 색으로 나타내 주고, 유전자의 위치에서 마우스를 클릭함으로써 해당되는 유전자의 서열을 볼 수 있다.

### 3.3 CGR 뷰어(Chaos Game Representation Viewer)

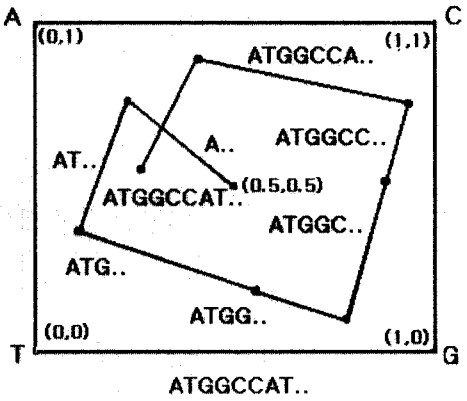
가시화 시스템에서는 DNA 서열 분석을 위한 가시화 방법 중 하나인 CGR(Chaos Game Represenation)을 제공한다. CGR은 DNA 서열에서 핵산이나 단백질 서열에서 아미노산과 같은 단위의 서열들을 처리하는 반복적인 매핑 기술로 연속 공간에서 위치의 좌표들을 결정한다. 위치의 분포는 유일하고 좌표로부터 서열을 복구할 수 있는 두 가지 성질을 갖는다. CGR을 이용하면 전체 거리(global distance)와 부분 유사도(local similarity)를 계산할 수 있고, 다변량(multivariate) 통계 분석 방법을 적용할 수 있다[1, 2, 8, 12].

CGR은 1990년에 Jeffrey가 유전체의 서열을 위한 scale-independent 표현으로 제안하였다[15]. 원래의 제안은 임의의 심벌 서열을 위해 확장되고 일반화되어, 단백질 서열과 같은 다른 생물학적 서열을 포함하게 되었다. CGR 공간은 어떤 길이의 모든 가능한 서열들이 가지는 유일한 위치를 가지는 연속된 참조 시스템이다. 모든 가능한 nucleotide 연속 체계는 연속 공간에서 인코딩되고, non-integer 순서를 수용하는 scale-independent Markov 모델의 새 일반화이다[15]. 유전체의 서열로 생성된 CGR 공간은 평면이고, 네 개의 가능한 nucleotide를 바이너리 정사각형의 정점으로 하는 영역의 안에 있게 된다. 길이 $n_g$인 서열 $g$의 각 nucleotide $g_i$의 CGR 위치 $CGR_i$는 포인터를 이전의 위치와 현재 바이너리 표현 사이의 거리 반으로 이동함으로써 계산된다. 바이너리 CGR 정점은 네 개의 nucleotide에 할당된다[1].

$$CGR_i = CGR_{i-1} - 0.5(CGR_{i-1} - g_i),$$
$$i = 1 \cdots n_g,\ CGR_0 = (0.5, 0.5)$$
$$g_i \in \{A = (0,0), C = (1,0), G = (1,1), T = (0,1)\}$$

결정된 염기의 수는 CGR 좌표의 해상도 함수이므로 nucleotide 서열은 CGR 좌표들로부터 복원될 수 있다.

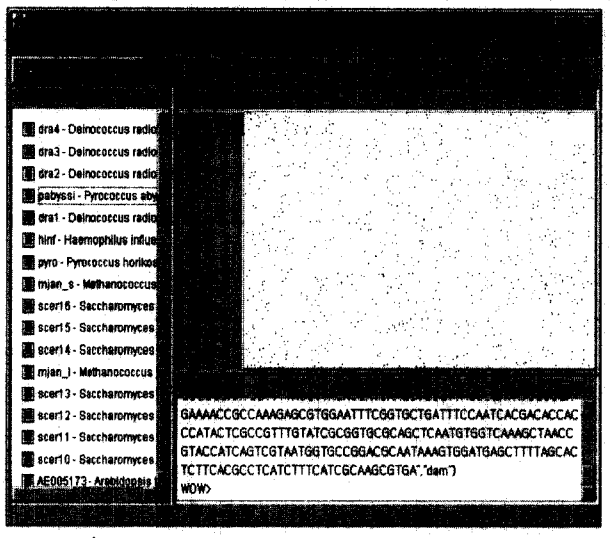
(그림 4)는 CGR을 그리는 방법에 대한 것이다. 그림에서 볼 수 있듯이 유전체의 서열로 생성된 CGR은 면적이 1이고 네 개의 가능한 핵산(nucleotide)을 정사각형의 정점으로 하는 영역의 안에 있게 된다. CGR의 시작은 중앙점(0.5, 0.5)에서 시작하여 현재점과 핵산이 있는 꼭지점와의 거리 반에 해당하는 점으로 이동함으로써 계산된다. 그림에서 처음이 A이므로 중앙점과 (0,1) 사이의 중간점으로 이동하고, 다음 점이 T이므로 현재점에서 (0,0) 사이의 중간점으로 이동한다. 계속적으로 반복하면 서열에 대한 CGR을 그릴 수 있다.



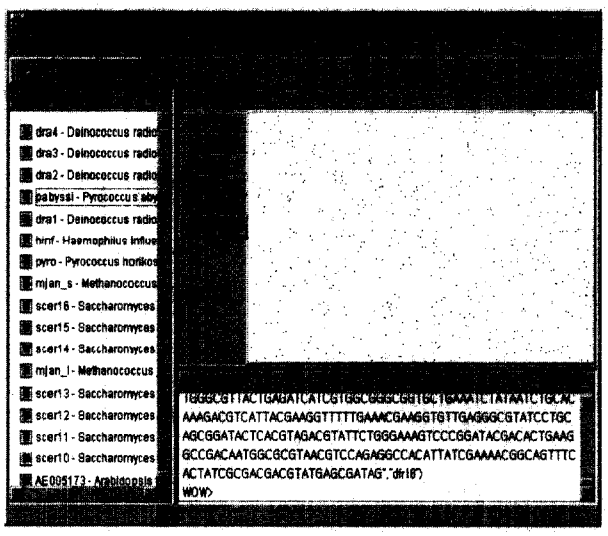

(그림 4) Vibrio cholerae의 유전자 tnpA의 처음 8개의 염기에 대한 CGR 이미지 : ATGGCCAT. 각 염기에 대한 좌표는 중앙점에서 시작하여 반복적으로 계산된다.

(그림 5)는 본 시스템에서 제공해주는 CGR 가시화 화면이다. 사용한 데이터는 Vibrio cholerae의 2가지 유전자 dam, dfr18이다. Dam과 dfr18은 서열의 길이가 비슷하지만, 각각의 유전자 서열에 따라서 CGR의 결과도 달라짐을 알

수 있다. 그런데 CGR 표면에 찍히는 점들은 각 서열이 무엇이냐에 따라서 정해지지만, 각 서열들의 CGR 위치간 거리는 서로간의 유사도에 따라 결정된다. 만약 두 서열의 마지막 염기가 같다면 마지막 두 점간의 거리는 0.5를 넘지 않을 것이다. 또한 서열의 마지막 두 염기가 같다면 마지막 점의 위치거리는 0.25를 넘지 않을 것이다. 이러한 점을 이용하여, 서열의 유사도 측정이 가능하다[1]. 각각의 유전자 서열에 따라 CGR의 결과도 달라짐을 알 수 있다.




(a) DNA adenine methylase(dam) 유전자

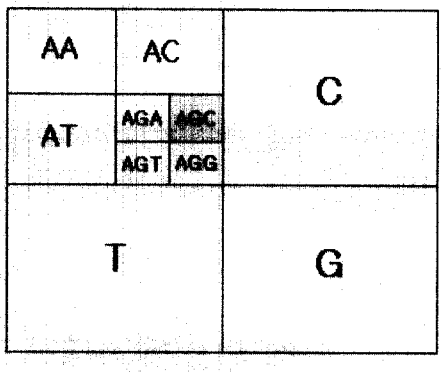



(b) dfr18 유전자

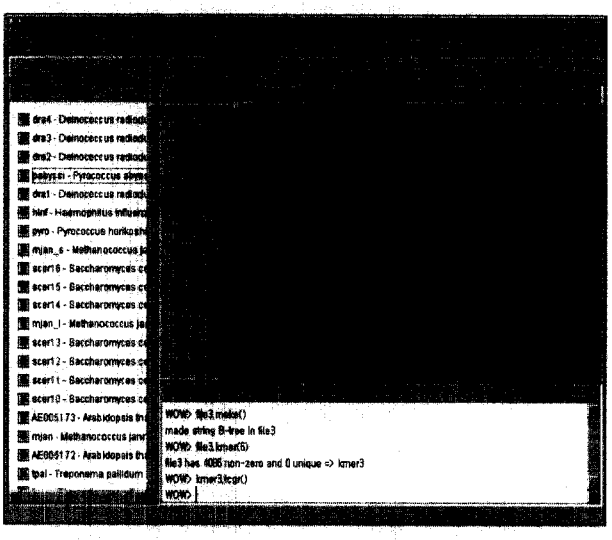
(그림 5) Vibrio cholerae의 두 유전자의 CGR

### 3.4 *k*-mer 뷰어(*k*-mer Viewer)

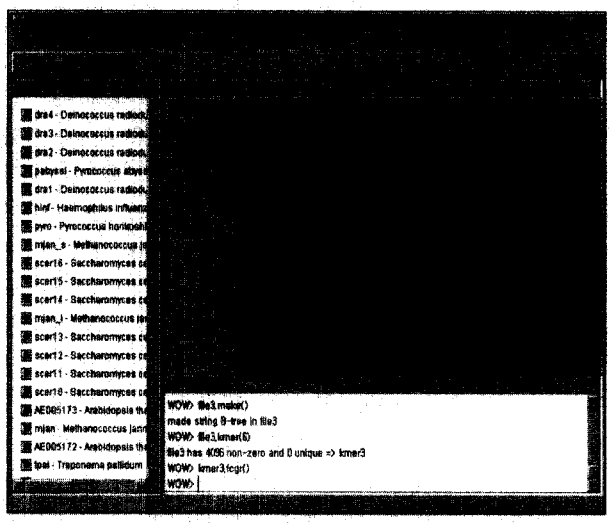
*k*-mer 분석의 결과를 가시화하는 것이 *k*-mer 뷰어이다. 하나의 사각형을 4등분하여 각각을 A, C, G, T의 영역으로 할당하고 각 사각형을 다시 4등분하여 AA, AC, AG, …, TT의 영역으로 할당한다. 이렇게 *k*번 계속 4등분해 나가면 *k*-mer의 영역을 만들 수 있다. 각 *k*-mer의 빈도는 색깔로 나타낸다. (그림 6)은 AGC의 위치를 나타낸 것이다(색이 칠해져 있는 부분).




(그림 6) *k*-mer graph에서 AGC의 위치(색이 칠해져 있는 부분)




(a) Mycoplasma genitalium G37




(b) Rhizobium sp. NGR234 plasmid pNGR234a

(그림 7) 6-mer에 대한 *k*-mer graph

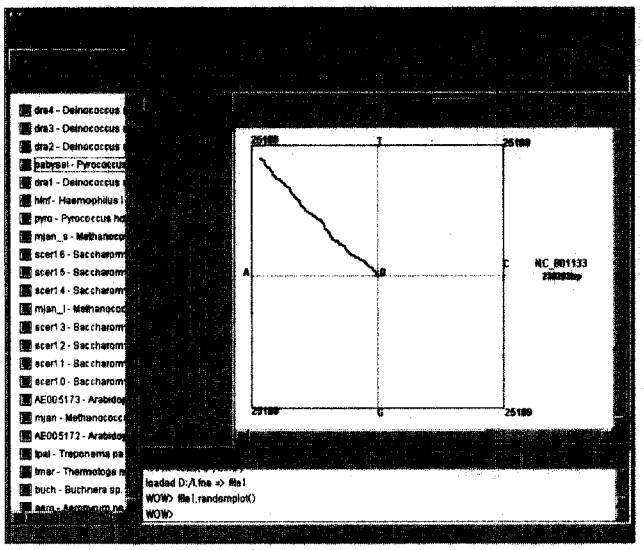
본 시스템에서는 입력으로 받아들인 유전체 서열의 $k$-mer를 구하고 가시화하는 뷰어를 제공한다. 먼저 빈도를 최대 빈도에 대해 정규화하고 그 값으로 색깔을 보간 한다. 빈도율을 7단계로 나누는데 각 단계는 검은색, 빨간색, 노란색, 녹색, 청록색, 파란색, 흰색이다. 단계 사이에 있는 것은 단계의 색으로 보간하여 나타낸다[6].

(그림 7)은 본 시스템에서 제공하는 $k$-mer graph의 화면이다. (a)는 Mycoplasma genitalium G37, (b)는 Rhizobium sp. NGR234 plasmid pNGR234a의 6-mer을 $k$-mer graph로 나타낸 것이다. 그림을 살펴보면 종별로 특이한 패턴이 나타남을 알 수 있다. 또한 동종에 대해서는 같은 패턴을 나타낸다.

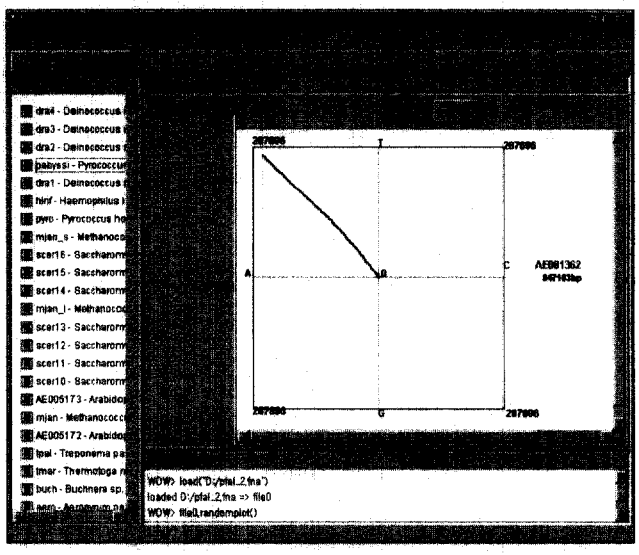
### 3.5 RWP 뷰어(Random Walk Plot Viewer)

RW(Random Walk)는 한 지점에서 시작하여, 매회 일정한 거리 d만큼 랜덤 방향으로 움직일 때 n회가 지난 후 현재의 위치가 출발 위치에서 거리 d와 d+rd 사이에 있을 확률을 구하는 문제이다. 랜덤 방향은 1차원 뿐만이 아니라 2차원과 3차원으로 확장할 수 있다.

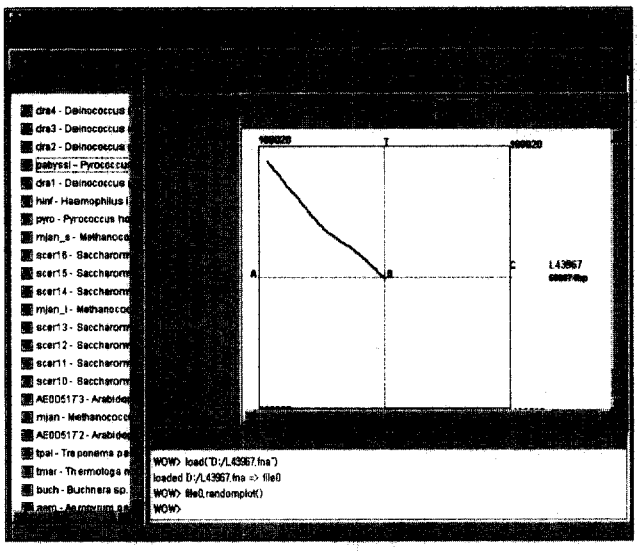
가시화 시스템에서는 RWP(Random Walk Plot)를 이용하여 유전체의 DNA 서열 모습을 표현해 준다[18]. RWP의 방향은 아데닌(A, Adenine)은 왼쪽으로, 티민(T ,Thymine)은 위로, 사이토신(C, Cytosine)은 오른쪽에, 구아닌(G, Guanine)은 아래쪽 방향으로 이동하게 하였다. 이렇게 그려진 유전체의 RWP는 각 유전체 서열들의 유사도나, 그 서열에서 어떠한 염기쌍이 많이 존재하는가와 같은 정보들을 보



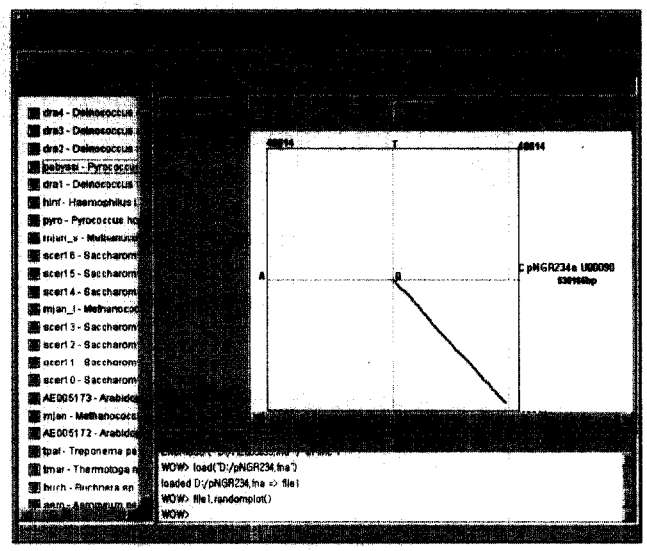
(a) Saccharomyces cerevisiae 염색체 I(2302036bp)



(b) Vibrio cholerae 염색체 I(9471036bp)



(c) Mycoplasma genitalium G37(580074bp)



(d) NGR234 plasmid pNGR234a(5361656b)

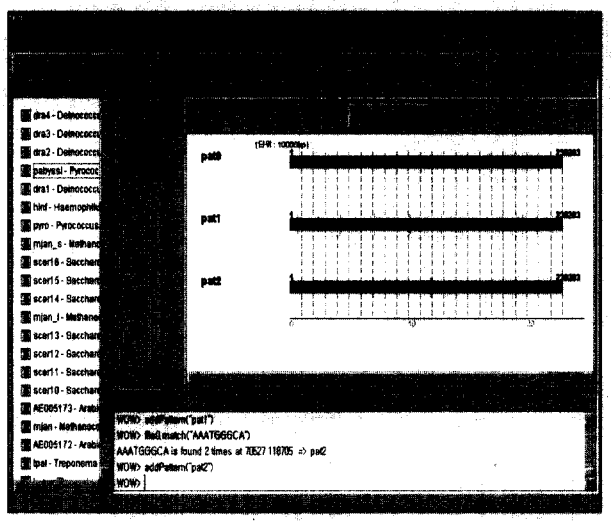
(그림 8) RWP(Random Walk Plot)

여준다. 서열의 특징 중 하나인 GC 양을 RWP를 이용하여 알 수 있다. 만약 AT가 많다면 매번 움직일 때마다 A와 T 쪽으로 많이 움직이기 때문에 왼쪽 윗부분에 그림이 그려질 것이다. 반면 GC가 많다면 오른쪽 아래쪽에 그림이 그려질 것이다.

(그림 8)의 (a)는 Saccharomyces cerevisiae 염색체 I(2302036bp), (b)는 Vibrio cholerae 염색체 I(9471036bp), (c)는 Mycoplasma genitalium G37(580074bp), (d)는 NGR234 plasmid pNGR234a(5361656b)의 RWP이다. 그림에서 (a), (b), (c)는 전체적인 모양이 서로 유사함을 알 수 있다. 반면, (d)는 다른 모습을 하고 있음을 알 수 있다. 또한 (a), (b), (c)의 서열에는 AT의 양이 많음을 알 수 있고, (d)에는 GC의 양이 많음을 알 수 있다. 그림에서 서로 비슷한 결과를 보였던, Saccharomyces cerevisiae 염색체 I와 Vibrio cholerae 염색체 I, Mycoplasma genitalium G37의 결과를 확대하여 살펴보면 모두 서열에 AT의 비율이 CG보다 많아 서로 비슷하게 그림이 그려져 있지만, RWP의 뻗어나가는 그림의 모양은 서로 다름을 알 수 있다. (a)는 (b)와 (c)에 비하여 곧게 그려져 있지 않다. 이것은 (a)의 서열에 AT 함량이 많기는 하지만, (b)나 (c) 보다 CG의 양이 많기 때문에 선이 곧게 그려져 있지 않은 것이다. 이처럼 RWP는 해당 유전체 서열의 특징을 반영하며, 유전체에 따라서 달리 나타나므로 유전체들 간의 비교에 이용할 수 있다.

### 3.6 맵 뷰어(Map Viewer)

서열 비교 뷰어는 하나 이상의 유전체들에서 어떤 서열을 검색하고, 그 서열이 어느 위치에서 나타나는 지를 비교해 볼 때 사용하는 뷰어이다. 이것을 통해서 여러 유전체에서 특정 서열들이 어떻게 분포하는지를 알아볼 수 있다. 이렇게 검색된 서열들을 가시화 시켜주는 시스템은 사용자로



(그림 9) 서열 비교 뷰어 : 여러 개의 유전체에 대해 특정 서열이분포하는 것을 비교할 수 있고 원형이나 선형 모양으로 볼 수 있다.

하여금 패턴들의 위치 정보를 텍스트로 보여주는 것보다쉽게 인식할 수 있게 해준다. 가시화 시스템에서 제공하는 패턴 검색 뷰어에는 사용자의 요구에 따라 전유전체 서열을 라인이나 원형으로 보여주고 확대 및 축소할 수 있다. (그림 9)는 세 가지 유전체에서 서열을 검색했을 때, 그 결과를 보여주는 것이다.

## 4. 워크벤치의 응용

워크벤치는 유전체 서열 분석을 위한 기본적인 질의를 제공하고, 이 질의들을 이용하여 프로그래밍할 수 있는 스크립트를 제공하므로 사용자들은 자신의 연구에 맞추어 질의문을 응용할 수 있다. 이 스크립트는 인터넷 표준 언어인 자바 스크립트와 유사하므로 배우기 쉽고 범용적이다. 이 스크립트를 사용하여 다양한 문제를 해결할 수 있는데 본 절에서는 결여 서열에 대해 조사한다.

사람과 쥐의 유전체 서열에는 CG가 잘 나타나지 않고, *Archeglobus fulgidus* 등과 같은 미생물 유전체 서열에는 CTAG 서열이 잘 나타나지 않는다[8, 22]. 이렇게 유전체 서열에서 잘 나타나지 않는 서열을 결여 서열(Avoided Sequence)이라 한다[14]. 결여 서열을 찾는 것은 $k$-mer 분석을 하여 빈도가 임계값(threshold) 이하인 $k$-mer를 찾으면 된다. 실험에서는 임계값을 0.002로 하였다. (알고리즘 4)은 결여 서열을 구하는 알고리즘이다.

| AvoidedKmer(genome, threshold) : 유전체 서열에 잘 나타나지 않는 서열 검색 |
|---|
| 입력 : genome : 유전체 서열<br>threshold : 결여 서열 판단 기준값 |
| 출력 : kmer : 결여 서열 조각 |

```
for ( k = 1 ; k < 20 ; k++ )
{
    temp = genome.kmer ( k ) ;
    for ( i = 0 ; i < temp.size ( ) ; i++ )
    {
        if ( temp.freq ( i ) / genome.length ( ) < threshold )
          kmer.add ( temp.seq ( i )) ;
    }
}
```

(알고리즘 4) 결여 서열을 구하는 알고리즘

<표 1>은 실험에 사용된 유전체들이고 NCBI(ftp : //ncbi.nlm.nih.gov/genbank/genomes)에서 다운로드 받을 수 있다. 각 유전체들은 분류학(taxonomy)에 따라 4개의 그룹으로 나누었다.

<표 2>는 <표 1>의 유전체에 대한 결여 서열 중 길이가 가장 작은 것들을 보여준다(Shortest Avoided Sequence). 여기에서 두 번째 열은 SAS의 길이이고 세 번째 열은 개수이다.

〈표 1〉 실험에 사용된 21종의 전유전체(whole genome)

| Genome | Abb. | Base Pair | Taxonomy | |
|---|---|---|---|---|
| Archaeoglobus fulgidus | Aful | 2178400 | Archaea | Euryarchaeota |
| Halobacterium sp. NRC-1 | Hbsp | 2014239 | | |
| Methanobacterium thermoautotrophicum | Mthe | 1751377 | | |
| Methanococcus jannaschii | Mjan | 1664970 | | |
| Pyrococcus abyssi | Paby | 1765118 | | |
| Thermoplasma acidophilum | Tacid | 1564906 | | |
| Mycobacterium leprae | Mlep | 3268203 | Bacteria | Firmicutes |
| Mycobacterium tuberculosis H37Rv | Mtub | 4411529 | | |
| Bacillus subtilis | Bsub | 4214814 | | |
| Clostridium acetobutylicum ATCC824 | Cace | 3940880 | | |
| Mycoplasma genitalium | Mgen | 580074 | | |
| Lactococcus lactis subsp. lactis | Llac | 2365589 | | |
| Streptococcus pneumoniae | Spneu | 2160837 | | |
| Caulobacter crescentus | Ccre | 4016947 | Bacteria | Proteobacteria |
| Sinorhizobium meliloti 1021 | Smel | 3654135 | | |
| Neisseria meningitidis MC58 | Nmen | 2272351 | | |
| Campylobacter jejuni | Cjej | 1641481 | | |
| Helicobacter pylori 26695 | Hpyl | 1667867 | | |
| Escherichia coli | Ecoli | 4639221 | | |
| Haemophilus influenzae Rd | Hinf | 1830138 | | |
| Vibrio cholerae chr. I | Vcho | 2961149 | | |
| Plasmodium falciparum chr. III | Pfal3 | 1060106 | Eukaryote | Alveolata |
| Saccharomyces cerevisiae chr. I | Scer1 | 230203 | | Fungi |
| Homo sapiens chromosome XXI | Hs21 | 34004148 | | Metazoa |
| Caenorhabditis elegans chr. I | Cel1 | 16183833 | | |
| Arabidopsis thaliana chr. II | Ath2 | 19647091 | | Viridiplantae |

<표 3>은 알려진 결여 서열들의 빈도 백분율을 보여주는데, 진한 회색은 0.05%이하이고, 보통 회색은 0.1%이하이고, 밝은 회색은 0.2%이하를 나타낸다. 실험에 사용된 유전체는 총 26개인데 CTAG가 결여 서열인 유전체는 23개로 가장 많고, CGCG는 17개, GTAC, GCGC, CCGG는 15개, GGCC는 14개, ACGT는 11개, TCGA는 9개, GATC는 7개 순으로 분포한다.

〈표 2〉 유전체 서열에서 0.1% 이하로 나타나는 서열 중 길이가 가장 작은 서열

| Abb. | Length | Number | Sequence |
|---|---|---|---|
| Aful | 4 | 2 | CGCG CTAG |
| Hbsp | 4 | 50 | AAAA AAAG AAAT AATA AATG AATT ACTA ATAA ATAC ATAG ATAT ATTA ATTC ATTG ATTT ··· |
| Mthe | 4 | 4 | CGCG CTAA CTAG TTAG |
| Mjan | 4 | 54 | ACCG ACGA ACGC ACGG ACGT AGCG ATCG CACG CCCG CCGA CCGC CCGG CCGT CGAA CGAC ··· |
| Paby | 4 | 2 | CGCG GCGC |
| Tacid | 4 | 1 | CTAG |
| Mlep | 5 | 627 | AAAAA AAAAC AAAAG AAAAT AAACA AAACC AAACG AAACT AAAGA AAAGC AAAGG AAAGT AAATA AAATC AAATG ··· |
| Mtub | 4 | 29 | AAAA AATA ACTA AGTA ATAA ATAT ATTA CTAA CTAG CTTA GTAA GTTA TAAA TAAC TAAG ... |
| Bsub | 4 | 1 | CTAG |
| Cace | 4 | 48 | ACCG ACGA ACGC ACGG ACGT AGCG ATCG CACG CCCC CCCG CCGA CCGC CCGG CCGT CGAC ··· |
| Mgen | 4 | 52 | ACCG ACGA ACGC ACGG ACGT AGGC ATCG CACG CCCG CCGA CCGC CCGG CCGT CCTC CGAA ··· |
| Llac | 4 | 9 | CCCC CCCG CCGG CGCG CGGG GATC GCGC GGCC GGGG |
| Spneu | 4 | 5 | CCCG CCGG CGCG CGGC CGGG |
| Ccre | 4 | 43 | AAAA AAAT AATA AATG AATT ACTA AGTA AGTG ATAA ATAC ATAT ATTA ATTC ATTT CACT ··· |
| Smel | 4 | 24 | ACTA AGTA ATAA ATTA CTAA CTAG CTTA GTAA GTAC GTTA TAAA TAAC TAAG TAAT TACA ··· |
| Nmen | 4 | 2 | CTAG GATC |
| Cjej | 4 | 44 | ACCG ACGG ACGT CACG CCAG CCCC CCCG CCGA CCGC CCGG CCGT CCTC CGAC CGAG CGCC ··· |
| Hpyl | 4 | 17 | ACGT ACTG CCGA CGAC CGGA CGTC GACC GACG GGAC GGCC GGTC GTAC GTCC GTCG TCCG ··· |
| Ecoli | 4 | 3 | CCTA CTAG TAGG |
| Hinf | 4 | 6 | CCGG CGGG GGAC GGAG GGCC GTCC |
| Vcho | 5 | 584 | AACCC AACCT AACGA AACGG AACGT AACTA AAGAC AAGGA AAGGG AAGTA AAGTC AATAG AATCT AATGT AATTA ··· |

## 5. 결 론

본 논문에서는 전유전체 서열(whole genome sequence)의 분석과 가시화를 위한 워크벤치를 개발하였다. 워크벤치는 대용량인 유전체 서열을 처리하기 위해서 외부 메모리 자료구조에 가장 좋은 B-트리와 서픽스 배열의 특성을 조

〈표 3〉 알려진 결여 서열의 빈도 백분율

| Abb. | ctag | acgt | gatc | gtac | tcga | gcgc | cgcg | ggcc | ccgg |
|---|---|---|---|---|---|---|---|---|---|
| Aful | | 0.269% | | | 0.419% | | | 0.256% | 0.249% |
| Hbsp | | 0.586% | 0.548% | 0.364% | 1.096% | 1.294% | 1.774% | 0.831% | 1.133% |
| Mthe | | | 0.368% | 0.263% | 0.224% | | | 0.499% | 0.415% |
| Mjan | | | | | | | | | |
| Paby | 0.325% | 0.322% | 0.234% | 0.257% | 0.359% | | | 0.360% | |
| Tacid | | 0.269% | 0.831% | 0.295% | 0.398% | 0.214% | | 0.433% | 0.352% |
| Mlep | 0.220% | 0.366% | 0.623% | 0.282% | 0.674% | 0.699% | 0.676% | 0.661% | 0.792% |
| Mtub | | 0.346% | 0.713% | 0.205% | 0.726% | 1.211% | 1.197% | 1.142% | 1.341% |
| Bsub | | 0.215% | 0.427% | | 0.260% | 0.293% | 0.201% | 0.225% | 0.345% |
| Cace | | | | 0.214% | | | | | |
| Mgen | | | 0.278% | | | | | | |
| Llact | | 0.204% | | | | | | | |
| spneu | 0.297% | | | | 0.237% | | | | |
| Ccre | | 0.222% | 0.958% | | 0.820% | 1.791% | 1.504% | 1.560% | 0.985% |
| Smel | | 0.227% | | | 0.983% | 1.356% | 1.207% | 0.874% | 1.116% |
| Nmen | | 0.210% | | 0.227% | 0.395% | 0.723% | 0.712% | | 0.426% |
| Cjej | 0.204% | | 0.216% | | | | | | |
| Hpyl | 0.259% | | 0.325% | | | 0.376% | 0.156% | | 0.104% |
| Ecoli | | 0.314% | 0.412% | 0.259% | 0.333% | 0.756% | 0.608% | 0.271% | 0.524% |
| Hinf | | 0.309% | 0.266% | | | | | | |
| Vcho1 | | 0.269% | 0.480% | 0.269% | 0.397% | 0.597% | 0.442% | 0.251% | |
| Pfal3 | | | | | | | | | |
| Scer1 | 0.202% | 0.259% | 0.280% | 0.308% | 0.293% | | | | |
| Hs21 | 0.251% | | 0.239% | | | | | 0.298% | |
| Cel1 | | | 0.254% | | 0.406% | | | | |
| Ath2 | 0.234% | | 0.359% | | 0.288% | | | | |
| Total | 23 | 11 | 7 | 15 | 9 | 15 | 17 | 14 | 15 |

합한 패턴 일치를 위한 자료 구조인 스트링 B-트리를 질의어 시스템의 엔진으로 사용한다. 이 시스템은 서열 분석을 위한 질의문 처리 부분과 가시화 부분으로 나뉘어 지고, 워크벤치의 장점은 다음과 같다.

- 유전체 서열 분석을 위해 스트링 B-트리를 사용하기 때문에 다른 색인 자료구조들보다 메모리를 적게 사용한다.
- 유전체 서열 분석을 위한 기본적인 질의를 지원함으로써 복잡하고 다양한 분석과 응용을 할 수 있다.
- 본 시스템은 자바 언어로 개발되어 플랫폼에 독립적이고, 기본 질의를 기반으로 프로그래밍할 수 있는 스크립트를 제공한다.
- 다양한 가시화 방법을 제공하여 유전체 서열 분석 결과를 직관적으로 인지할 수 있도록 해준다.

본 연구에서 개발한 워크벤치를 이용하여 다양한 유전체 서열 분석 작업을 할 수 있는데, 본 논문에서는 결여 서열(Avoided Sequence)을 분석하였다. 종별로 한번 나타나는 oligo-nucleotide가 다를 수 있으므로, 이것은 oligo-chip을 제작할 때 probe로서 사용될 수 있다. oligo-chip은 특정 생물종이나 질병을 진단할 수 있다. 한번도 나타나지 않는 oligo-nucleotide는 단백질로 만들어지지 않는 아미노산이므로, 이 아미노산을 가지는 단백질이 특정 종에 존재하지 않음을 알 수 있고, 진화적 관점에서 볼 때 돌연변이가 잘 일어나는 oligo-nucleotide로 간주할 수 있다. 또한 짧은 길이의 oligo-nucleotide의 빈도수가 각 생물종에게 있어 종 특이적인 패턴을 나타내므로 $k$-mer 분석이 매우 중요하다. 결여 서열 분석과 $k$-mer 분석은 4절에서 볼 수 있듯이 워크벤치에서 제공하는 질의어와 스크립트를 이용하여 쉽게 조사할 수 있다.

## 참 고 문 헌

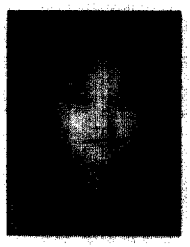
[1] J. S. Almeidal, J. A. Carric, A. Maretzek, P. A. Noble, and M. Fletcher, Analysis of genomic sequences by Chaos Game Representation, Bioinformatics, Vol.17, No.5, pp.429-437, 2001.

[2] S. Basu, A. Pam, and J. Das, Chaos game representation

of protein, J. Mol. Graphics Mod., Vol.15, pp.279-289, 1997.
[3] B. E. Blaisdell, A. M. Campbell, and S. Karlin, Similarities and dissimilarities of phage genomes, Proc. Natl. Acad. Sci., Vol.93, pp.5854-5859, 1996.
[4] C. Burge, A. M. Campbell, and S. Karlin, Over-and under-representation of short oligonucleotides in DNA sequences, Proc. Natl. Acad. Sci., Vol.89, pp.1358-1362, 1992.
[5] J. H. Choi and H. G. Cho, An analysis for whole genomic sequence using string B-tree, The KIPS Trans., Vol.8-A, No.3, pp.253-260, 2001.
[6] J. H. Choi, S. K. Lee, S. B. Lee, Y. J. Kim, H. G. Cho, and K. W. Kim, Analysis of genome by visualization of genomic signature, Korean J. Genetics, Vol.24, No.1, 2002.
[7] A. Compell, J. Mrzek, and S. Karlin, Genome signature comparisons among prokaryote, plasmid, and mitochondrial DNA, Proc. Natl. Acad. Sci., Vol.96, pp.9184-9189, 1999.
[8] P. J. Deschavanne, A. Giron, J. Vilain, G. Fagot, and B. Fertil, Genomic signature : Characterization and classification of species assessed by chaos game representation of sequences, Mol. Biol. Evol., Vol.16, pp.1391-1399, 1999.
[9] DNASpace, http://www.hitachi-sk.co.jp, Hitachi Co.
[10] P. Ferragina and R. Grossi, The string B-tree : A new data structure for string search in external memory and its application, Journal of ACM, Vo.46, No.2, pp.236-280, 1999.
[11] GenoMax, http://www.informaxinc.com, InforMax Co.
[12] N. Goldman, Nucleotide, dinucleotide, and trinucleodtide frequencies explain patterns observed in chaos game respresentations of DNA sequences, Nuclear Acids Res., Vol. 21, pp.2487-2491, 1993.
[13] D. Gusfield, Algorithms on strings, trees, and sequences, Cambridge Univ. Press, 1997.
[14] B. L. Hao, Fractals from genome exact solutions of a biology-inspired problem, Physica A., Vol.282, pp.225-246, 2000.
[15] H. J. Jeffery, Chaos game representation of gene structure, Nucleic Acids Res., Vol.18, pp.2163-2170, 1990.
[16] S. Karlin and I. Ladunga, Comparisons of eukaryotic genomic sequences, Proc. Natl. Acad. Sci., Vol.91, pp.12832-12836, 1994.
[17] S. Karlin and J. Mrzek, Compositional differences within and between eukaryotic genomes, Proc. Natl. Acad. Sci., Vol.94, pp.10227-10232, 1997.
[18] P. M. Leong and S. Morgenthaler, Random walk and gap plots of DNA sequences, SO Comput-Applic-Biosci, Vol. 22, No.5, pp.935-948, 1993.
[19] M. C. MacLeod, D. A. Johnston, M. LaBate, and R. A. White, The probability of occurrence of oligomer motifs in the human genome and genomic microheterogeneity, J. Theor. Biol., Vol.181, No.4, pp.311-318, 1996.
[20] U. Manber and G. Myers, Suffix arrays : A new method for on-line string searches, SIAM J. on Comp., Vol.22, No.5, pp.935-948, 1993.
[21] G. J. Phillips, J. Arnold, and R. Ivarie, Mono-through hexanucleotide composition of the *Escherichia coli* genome : a Markov chain analysis, Nucleic Acid Res., Vol.15, pp.2611-2626, 1987.
[22] N. J. Robinson, P. J. Robinson, A. Gupta, A. J. Bleasby, B. A. Whitton, and A. P. Morby, Singular overrepresentation of an octameric palindrome, HIP1, in DNA from many cyanobacteria, Nucleic Acid Res., Vol.23, pp.729-735, 1995.
[23] VectorNTI, http://www.informaxinc.com, InforMax Co.

**최 정 현**

e-mail : jhchoi@pearl.cs.pusan.ac.kr
1995년 부산대학교 물리학과 졸업(학사)
2000년 부산대학교 대학원 전자계산학과 졸업(이학석사)
2000년~현재 부산대학교 대학원 전자계산학과 박사과정
관심분야 : 스트링 매칭, 생물정보학, 정보 가시화

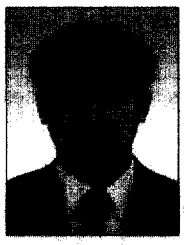


**진 희 정**

e-mail : hjjin@pearl.cs.pusan.ac.kr
2000년 부산대학교 전자계산학과 졸업(학사)
2002년 부산대학교 전자계산학과 졸업(이학석사)
2002년~현재 국립보건원 유전체연구소 역학정보실 생물정보학팀 선임연구원
관심분야 : 생물정보학, 알고리즘이론

**김 철 민**

e-mail : kimcm@pusan.ac.kr
1988년 부산대학교 의과대학 의학과 졸업(의학사)
1990년 부산대학교 대학원 의학과 졸업(의학석사)
1993년 부산대학교 대학원 의학과 졸업(의학박사)
1998년~현재 부산대학교 의과대학 의학과 조교수(생화학전공)
관심분야 : 인체유전학, 분자진단학, 생명의료정보학, 비교유전체학



**장 철 훈**

e-mail : kimcm@pusan.ac.kr

1986년 부산대학교 의과대학 의학과 졸업 (의학사)

1989년 부산대학교 대학원 의학과 졸업 (의학석사)

1995년 부산대학교 대학원 의학과 졸업 (의학박사)

1998년~현재 부산대학교 의과대학 의학과 기금조교수(임상병리학전공)

관심분야 : 임상미생물학, 분자진단학, 의료정보학

**조 환 규**

e-mail : hgcho@pusan.ac.kr

1984년 서울대학교 계산통계학과 졸업 (학사)

1986년 한국과학기술원 전자계산학과 졸업 (공학석사)

1990년 한국과학기술원 전자계산학과 졸업 (공학박사)

1990년~현재 부산대학교 전기전자정보컴퓨터공학부 교수

관심분야 : 그래프 이론, 생물정보학, 그래픽스