

## BLAST/FASTA를 활용한 미생물 유전체 비교용 도구의 개발

태홍석<sup>1,2</sup> · 이대상<sup>2</sup> · 박 완<sup>1</sup> · 박기정<sup>2,\*</sup>

<sup>1</sup>경북대학교 자연과학대학 미생물학과, <sup>2</sup>(주)스몰소프트 정보기술연구소

미생물 유전체 프로젝트의 결과인 유전체 서열에 대해, 비교 유전체 분석을 수행할 수 있는 분석 도구인 GComp를 개발하였다. 이 도구는 국부 상동성 계산을 BLAST나 FASTA를 사용하여 수행한 후에 그 결과를 받아들여, 상동성을 보이는 부분을 분석하고 위치 파악 및 연결한 뒤, 두 유전체간의 상동성 정도를 일목요연하게 보여줄 수 있는 테이블과 파일들을 생성한다. 한편, 그 결과를 그래픽으로 표시하고 전체를 살펴볼 수 있는 인터페이스 기능을 구현하였다. 시험 데이터로 기존의 미생물 유전체 서열을 상대로 분석하면서, 유전체 서열의 핵산 및 단백질 수준에서의 비교분석 결과를 통해 두 유전체에 대한 비교 정보를 효과적으로 확인할 수 있었고, 보다 다양한 분석을 위한 도구 개발의 기준을 설정할 수 있었다.

**Key words** □ BLAST, comparative genomics, FASTA, GComp, homology

1995년에 *Haemophilus influenzae* (9)의 완전한 유전체 서열이 공개된 후, 생명체 유전자 전체의 서열을 밝혀내자는 유전체 프로젝트가 활발히 진행되고 있다. 이에 따라 유전자 상호관계의 연구가 가능해지면서, 개별 유전자를 연구하던 유전학에서 유전체 자체를 연구하는 유전체학(genomics)이라는 학문으로 연구초점이 빠르게 움직이고 있으며, 그 연구 방법으로 유전체 수준에서 유전자의 기능 정보를 분석하고 비교하는 방법이 일반화되어 가고 있다.

한 유전체에 대한 분석 결과는 그 유전체에 포함된 유전자의 위치와 예측되는 기능들의 집합으로 표시된다. 이러한 유전체에 대한 정보는 주로 유전자 집합의 기능에 대한 위치별 리스트(5)나 gene ontology에 관한 데이터베이스의 클러스터별로 정리하여 표시(21,23)하거나, 가시화 도구(10)를 사용하여 원형의 유전체 지도 혹은, 선형의 유전체 지도로 표시한다. 현재로서는 이러한 유전체 분석 결과는 해당 홈페이지를 통해 인터넷에서 제한된 정보로만 살펴볼 수 있는 정도인데, 이러한 정보를 생성하거나 보다 효율적으로 이들 정보를 사용하기 위해서는 고가의 상용도구를 사용해야만 하고, 이러한 상용 프로그램들도 현재는 대부분 개발 단계에 있다.

### 재료 및 방법

#### 비교 유전체 방법

유전체 프로젝트의 중요한 목적이자 방법 중의 하나는, 대상이 되는 생명체의 유전체를 기존의 다른 유전체와 비교하여, 관련 유전자의 종류와 유사정도 등을 통해 유전자의 기능에 대한 포

괄적인 정보를 구하는 것(24)으로, 일반적으로 비교 유전체학(comparative genomics) (18,22)이라고 한다. 하나의 유전체에 대한 자체 분석이 완료되거나 진행 중일 때, 일반적으로 생물학자들이 가장 원하는 분석요구 중의 하나가 바로 유사하거나 관심 있는 다른 생명체와의 유전체 정보 비교분석이다. 유전체 정보를 비교하는 것은, 유전체에 대한 정보를 표시하는 방법의 표준화를 필요로 하는 작업이나, 현재 유전체 정보의 내용과 표시 방법이 임의적이므로, 비교 유전체 분석의 내용과 방법은 대단히 유동적일 수밖에 없다.

유전체를 유전체 핵산서열로 비교하는 방법과 유전체 상의 단백질 서열로 비교하는 것을 가장 일반적으로 생각해 볼 수 있는데, 실제로 그러한 방법들이 두 유전체의 비교를 위해 주로 사용되고 있다(3,4).

#### 유전체 비교 프로그램

현재 가장 많이 알려져 있는 도구로 TIGR에서 개발한 MUMmer (7,8)가 있으며, 핵산 비교를 위한 NUCmer와 단백질 비교를 위한 PROmer로 구성되어 있다. 단백질 비교는 예측된 단백질 서열에 대한 것이 아닌, 임의로 6개 프레임으로 변환한 서열에 대한 것이므로 유전자 예측 프로그램이나 그 기능과는 무관하다. 이 프로그램들의 장점으로 주장되는 것은 빠른 처리 속도인데, 이는 서열비교 알고리즘으로 crossmatch와 같은 방식을 사용하고 속도 개선을 위해 suffix tree (11)를 사용했기 때문이다. Suffix tree 알고리즘은 길이 n의 문자열이 있을 때 n개의 suffix S(i) (i번째에서 n번째까지의 서열)를 트리의 형태로 표현하여 문자열 검색에서 빠른 속도를 나타내도록 구성된 알고리즘이다. 한편, MUMmer 프로그램들의 처리 결과를 가시화하여 나타내기 위해 별도의 프로그램인 DisplayMUMs가 개발되었다. 이 프로그램들의 실행환경은 UNIX이며, 현재 Windows 환경에서의

\*To whom correspondence should be addressed.  
Tel: 042-864-2524, Fax: 042-866-9241  
E-mail: kjpark@smallsoft.co.kr

**Table 1.** Comparison of the features of genome alignment programs

	GComp	MUMmer	PipMaker	Vista
Integration of visualization function	Yes	No	Yes	Yes
Sensitivity of local alignment	High	Low	High	Low
Running Environment	Windows	Unix	Web	Web
Portability/User accessibility	Good	portable	Accessible via web	Accessible via web

버전은 개발되지 않았다.

또 다른 유전체 비교 프로그램으로 PipMaker (19)라는 프로그램이 있는데, 이 프로그램은 두 유전체의 염기 서열에 유전자 예측 부위와 반복 염기 서열을 별도로 표시를 한 후 BLAST를 수행한다. BLAST를 수행한 결과를 분석하고 정렬하여 percent identity plot (pip)와 유사한 정도를 화면에 출력하고 이와 함께 각 유전자의 위치를 같이 표시해 주고 있다. PipMaker는 인터넷 상의 웹 서버를 통해서만 서비스를 제공하는데, 화면출력은 PDF 파일로 만들어서 사용자에게 보내어준다.

PipMaker와 비슷한 출력을 보여주는 프로그램으로 vista (15)라는 프로그램이 있다. vista는 PipMaker와는 다르게 BLAST의 결과 대신 GLASS라는 전역 정렬(global alignment)을 수행하는 도구의 출력 결과와 Sanger Centre의 GFF 포맷인 주석(annotation) 파일을 분석해서 유전체 사이의 정렬된 모습을 PDF 파일로 만들어 준다. vista는 두 유전체 사이의 정렬된 모습뿐만 아니라 동시에 여러 유전체를 정렬한 모습을 한 화면에 보여줄 수 있다.

이처럼, 비교 유전체를 위한 분석 프로그램들은, 대체로 유전체 정렬(genome alignment)을 목표로 하여 그 방법과 분석 결과의 표현을 위한 방법들이 현재 개발되고 있다. 현재, 이들 프로그램은 초기 단계로서 유전체 연구자들의 요구사항이 정확히 반영되지 않으며, 알고리즘이나 인터페이스를 포함한 프로그램 구현의 다양한 면에서 지속적인 개발을 필요로 한다(Table 1). 한편, 이들 프로그램은 기존 유전체의 분석 결과를 활용하고 유전체 수준에서 정보를 활용하기 위한 주요한 데이터 마이닝 방법의 하나로서 가치가 매우 크므로, 학문적으로나 상업적인 목적에서 개발의 필요성이 큰 분야이다. 현재의 이들 프로그램들이 사용자 환경이나 인터페이스 및 계산 모듈 등에서 대단히 많은 제한이나 약점이 있으므로, 보다 나은 인터페이스와 알고리즘 개발 및 비교 유전체 방법의 개발 등을 통해 이들을 개발할 필요성과 여지가 대단히 크다고 할 수 있다.

### 서열 정렬 알고리즘

두 유전체를 비교하기 위해서는 서열 정렬(sequence alignment) 알고리즘을 사용해서 두 서열 간의 상동성을 측정해야 하는데, 서열 정렬은 서열 간의 상관관계를 보여주기 위해, 특히 상동성(homology)을 나타내기 위해 핵산이나 단백질의 서열을 정렬하는 것을 말한다.

일반적으로 비교할 서열의 길이 범위에 따라서 전역 정렬(global alignment)과 국부 정렬(local alignment)로 분류하는데,

유전체 정렬에서는 다양한 재조합과 변이에 의한 유전체 서열의 재정렬(rearrangement)을 전역 정렬로 표현하기 어렵기 때문에, 국부적으로 상동성을 가지는 지역을 검출하고 이들의 연결 관계를 분석하여 전체적인 유전체 상에서의 정렬을 재구성하는 방법을 사용해야 한다.

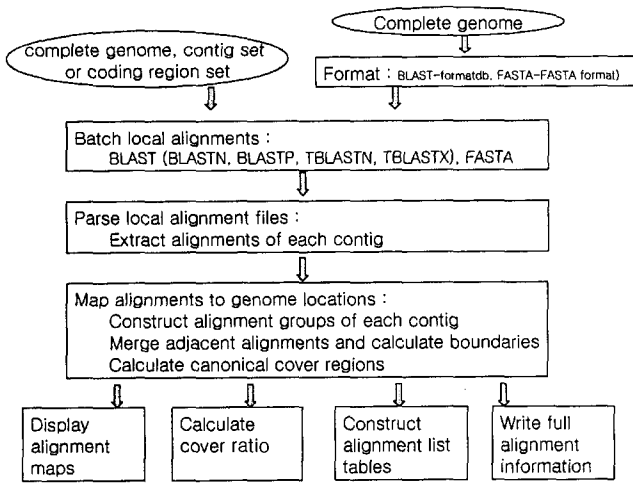
국부 서열 정렬 알고리즘으로 가장 광범위하게 사용되는 알고리즘은 Smith-Waterman 알고리즘, BLAST (1,2) 및 FASTA (16,18) 등이 있다. Smith-Waterman 알고리즘은 dynamic programming을 이용하여 전체 서열에서 유사성 검색을 수행하므로, FASTA와 BLAST에 비해 계산상 좀 더 정확한 검색결과를 얻을 수 있으나, 생물학적인 패턴의 국부 정렬에서는 오히려 BLAST나 FASTA보다 비효율적이고, 검색시간이 엄청나게 오래 걸린다는 단점이 있다. FASTA와 BLAST는 임의의 서열과 유사성을 가진 국부 서열을 서열 데이터베이스로부터 찾는 프로그램으로 일반적으로 상동성 검색을 위해 사용하는 프로그램이다. BLAST는 FASTA와는 달리 별도의 양식으로 미리 전처리된 검색 데이터베이스 파일을 필요로 한다. 두 가지 프로그램에서 사용하는 알고리즘은, 각각 옵션으로 입력한 통계치나 상동성 값을 필터로 하여 국부 상동성을 보이는 서열 부분을 계산 및 검색해 준다. 한편, 두 프로그램은 모두 gap을 포함한 정렬을 검색해 주므로, 유전체 정렬 계산을 위해 사용하는 부분 서열 정렬의 모듈 프로그램으로 적합하게 활용할 수 있다.

본 논문에서는, 미생물 유전체 프로젝트의 결과인 유전체 서열을 입력받아, 두 유전체를 핵산이나 단백질 서열의 수준에서 비교하는 비교 유전체 계산과 그 인터페이스를 포함한 기능을 하나의 도구로 통합하여 수행하는 프로그램으로 GComp를 개발하였다. 생물학 연구자의 일반적인 실험환경을 고려하여 VisualC++을 사용하여 윈도우즈상의 실행 프로그램으로 개발하였으며, 기존의 동일 목적 프로그램들의 기능을 최대한 반영하도록 설계하고 구현하였다. 상동성 비교의 효율을 위해 MUMmer에서 사용하는 알고리즘 대신 보다 효율적인 BLAST와 FASTA 프로그램을 활용하여, 그 결과를 비교에 반영하였다.

이 프로그램의 시험을 통해 여러 미생물 유전체에 대한 비교를 수행하여, 핵산과 단백질 수준의 비교에 대한 뚜렷한 차이를 볼 수 있었으며, 두 가지 비교에서 모두 일반적으로 예상할 수 있는 것보다 훨씬 낮은 수준의 유사성이 미생물의 유전체에서 보이는 것을 알 수 있었다.

### 시스템 구현과 방법

두 개의 유전체 서열을 국부 상동성 수준으로 비교하기 위해



**Fig. 1.** Overview of GComp. BLAST or FASTA alignment method is used as a local alignment module for whole genome comparison. A complete genome sequence is used as a DB sequence, and a complete genome sequence, contig set or coding region set is input as query or queries. GComp parses output files of BLAST and FASTA, and arranges the alignments on the DB genome sequence through the mapping algorithm. GComp displays mapped alignments and computes the cover ratio of aligned region against the DB genome region.

BLAST나 FASTA를 실행한다. BLAST나 FASTA에 의한 상동성 분석 결과를 파싱하여, 일정 수준 이상의 유사성을 보이는 것을 상동성으로 판정하여, 이들에 대한 위치와 표시 리스트를 작성한다. 작성된 리스트를 그래프로 표현한다(Fig. 1).

**국부 상동성 비교**

두 유전체를 전체 유전체 서열 차원에서 비교하기 위해 BLAST를 이용한다.

두 개의 유전체 서열 중 완성된 유전체 서열을 가진 하나의 유전체 서열에 대해 BLAST의 formatdb를 사용해서 BLAST DB로 구축하고, 다른 하나의 유전체 서열의 완성된 유전체 서열 또는 contig들을 질의어로 해서 BLASTN 계산을 실시한다(전체 유전체 서열을 입력으로 하는 경우에도 특정 길이 이하로 분할하여 질의어를 구성하므로, 이하 contig 서열들을 입력으로 하는 경우로 설명한다). 단백질 수준의 비교를 위해서 TBLASTX를 사용한다. 두 가지의 경우 별도의 파싱 프로그램을 사용한다. 한편, 상동성 비교 알고리즘에 대한 분석 결과를 비교하기 위해 BLAST와는 다른 알고리즘을 사용하는 프로그램으로 FASTA를 사용하여 국부 상동성 분석을 수행할 수 있다. 이 경우에도, 두 개의 유전체 서열 중 완성된 유전체 서열을 가진 하나의 유전체 서열을 library로 설정하고, 다른 유전체의 완성된 유전체 서열 또는 contig들을 질의어로 해서 FASTA 계산을 실시한다. FASTA의 경우에는 현재, 질의어 서열의 크기가 20 Kb 정도로 제한되어 있어 BLAST와의 계산 결과 비교용으로만 사용하였다.

FASTA 결과의 대한 파싱을 위해 FASTA 결과 분석용 파싱 프로그램을 별도로 구현하였다.

**위치 연결 알고리즘**

정렬 프로그램에 따라 파싱 부분은 다소 차이가 있지만, 처리 방법은 유사하다. 즉, 상동성 정렬 파일로부터 유의한 상동성의 정렬 리스트를 만들고, 이로부터 유전체 상의 위치로 연결(mapping)하는 방법을 사용한다. BLASTN의 검색 결과를 처리하는 경우를 예로 상동성 리스트 작성 과정을 설명한다. Fig. 2는 상동성 정렬 출력 파일을 파싱하여, 특정 유사성 이상을 나타내는 (E-value로 특정 cutoff value 이하) 두 서열 사이의 모든 정렬들을 링크드 리스트(linked list)에 연결한 후 각 정렬을 대상 유전체 서열의 해당 위치로 연결(mapping) 하는 알고리즘을 설명하고 있다.

Contig들이 질의어로 입력되면 각 contig 별로 정렬의 링크드 리스트를 따로 작성한다. 하나의 contig가 유전체의 여러 부분과 정렬되는 경우를 처리하기 위해, 정렬을 그룹으로 묶어서 처리하였다. 하나의 contig에 대해 유전체 내의 특정 거리 내의 유사한 위치에 있으면서 같은 방향을 갖는 정렬들의 집합을 하나의 그룹이라고 하였다. 여기서 특정 거리 값은 option으로 설정할 수도 있지만, 현재는 각 contig의 길이로 두고 처리하도록 하였다.

각 contig의 상동성 정렬 출력 파일을 파싱하여, 해당 contig에서 E-value가 가장 낮은 정렬을 선택하고 그 정렬의 유전체 서열상의 위치를 기준으로 앞 방향과 뒤 방향 각각 contig의 길이 범위 내에 위치하는 정렬들의 집합을 하나의 그룹으로 포함시키면서 링크드 리스트에 연결한다. 이때, 각 정렬의 유전체 상의 위치에 start position과 end position의 두 node를 생성하고, 이들을 소팅하면서 링크드 리스트를 생성한다. 즉, 정렬을 그룹에 포함시킬 때마다 정렬의 start position과 end position인 두 node들은 이 링크드 리스트의 node들을 순차적으로 검색해서 알맞은 위치로 끼어 들어간다. 그룹 형성에서 그룹에 포함되는지 여부를 결정할 때는, 그룹마다 하나의 개시 정렬(starting alignment)을 두고 이 정렬의 위치와의 거리를 기준으로 사용한다. 첫 번째 그룹의 개시 정렬은 그 contig에서 발생한 정렬 중에서 가장 E-value가 낮은 정렬이 되고, 다른 그룹의 개시 정렬은 그 이전 그룹들에 포함 되지 않으면서 정렬 E-value가 cutoff value 이하로 최소인 정렬이 선택된다. 이 기준에 만족하는 정렬이 없을 경우에는 더 이상의 그룹은 형성되지 않는다. 한 contig에서 형성할 수 있는 그룹은 default로 3 개로 두었고, option으로 20 개까지로 설정할 수 있다. E-value의 cutoff value는 option으로 설정할 수 있다.

이러한 처리 결과로, 각 contig에 대해서 한 개 이상의 그룹이 생성되면, 각 그룹은 정렬의 위치에 따라 sorting된 정렬 링크드 리스트로 구성된다.

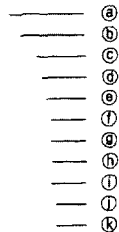
**유전체 정렬 가시화 표현**

각 그룹의 정렬 리스트에 대해, 순차적으로 정렬에 대한 start

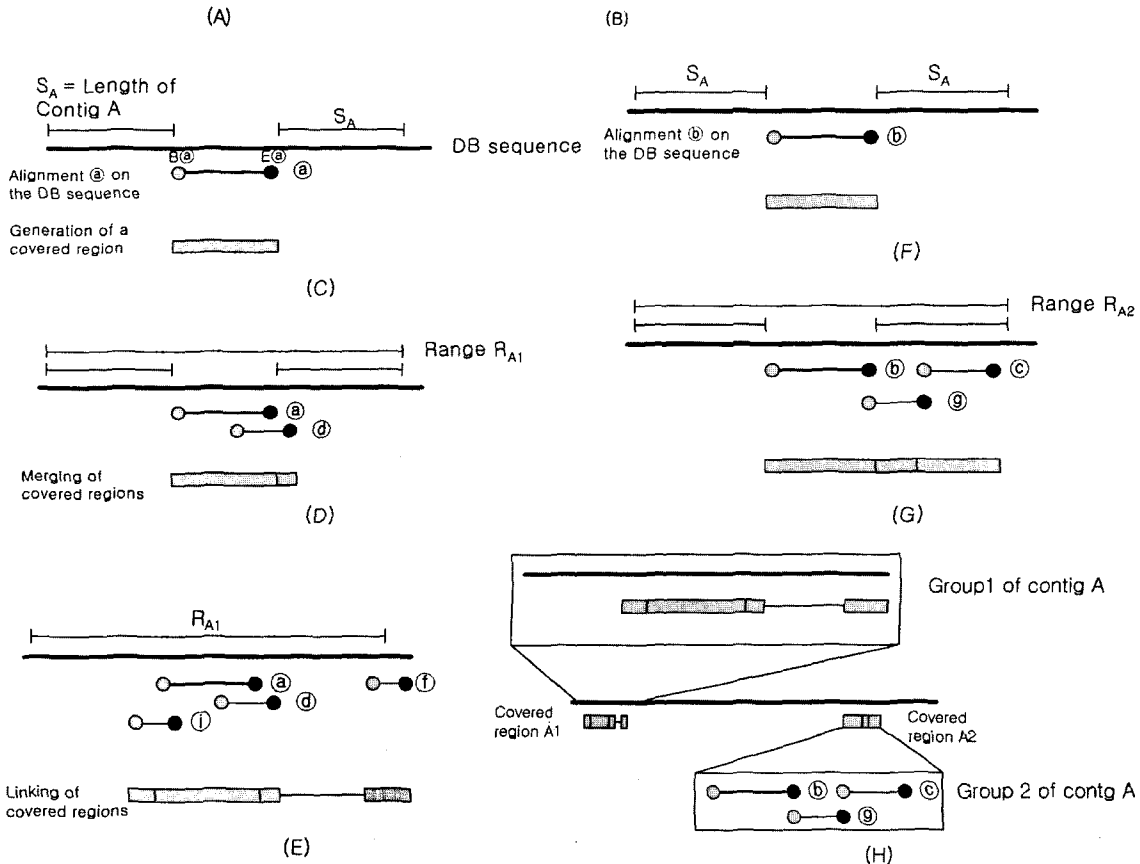
Contig A (Length : 5000bp) Alignment list

Id	E-value	length	Offset_in_DB	Group
Ⓐ	0.0	4300	15600	1
Ⓑ	2e-08	3500	54300	2
Ⓒ	0.001	1200	57830	2
Ⓓ	0.01	300	19850	1
Ⓔ	0.05	240	625	1
⓫	0.07	90	23500	1
⓬	0.41	55	57790	2
⓭	0.41	55	123500	2
⓮	0.23	32	15580	1
⓯	1.6	20	354320	1
⓰	1.6	20	524	1

Contig A alignment list



All alignments are added to the linked list for Contig A



**Fig. 2.** Mapping algorithm. (A): The list of alignment of contig A to the DB sequence. The alignment groups (as shown at the last column) for the contig are constructed from the alignment list as the result of the mapping algorithm. (B): The relative length of contig A alignments, (C): The cutoff E-value of alignment to make a group is 0.001. (D): Because the E-value of alignment Ⓐ is the lowest and is also lower than cutoff value, the alignment Ⓐ is selected as the starting alignment of group 1 (the alignment positions of DB are from B Ⓐ to E Ⓐ).  $R_{A1}$ , the position range of group 1 of contig A, are set as  $B^{Ⓐ}-S_A$  to  $E^{Ⓐ}+S_A$ . All alignments, which are placed in the range  $R_{A1}$  and overlapped with the current alignments, are added to the group 1 list and their covered regions are merged. (E): All alignments, which are placed in the range  $R_{A1}$  and not overlapped with the current alignments, are added to the group 1 list, and their covered regions are linked. (F): When there is no additional alignment for group 1, the alignment Ⓑ with the lowest E-value lower than the cutoff E-value among the remained alignments is selected as the starting alignment of group 2. (G): The steps like (D) to (E) are iterated for group 2. (H): The steps like (F) to (G) are iterated until no other group is found. And the final covered regions are drawn.

position과 end position 사이의 부분에 대해 유전체 서열과 정렬되는 부분(covered)으로서 직사각형으로 그 영역을 표시하고, 그룹의 범위 내에서 어떠한 정렬이 나타나지 않는 부분은 정렬되지 않는 부분(covered)으로서 비어있는 것으로 표시한다. covered 부분이 여러 정렬들로 연속되는 경우에는 인접한 직사각형의 병합된 형태로 표시하는데, 인접한 직사각형이 겹치는 경우

에는 E-value가 낮은 정렬의 위치로서 경계부분을 결정하여 표시(display)한다.

각 contig에 대해 정렬의 리스트를 별도의 윈도우를 통해 출력할 수 있는 기능을 구현하였다. 한편, 전체 contig 들에 대한 리스트를 하나의 텍스트 파일을 통해 출력할 수 있는 기능도 구현하였다.

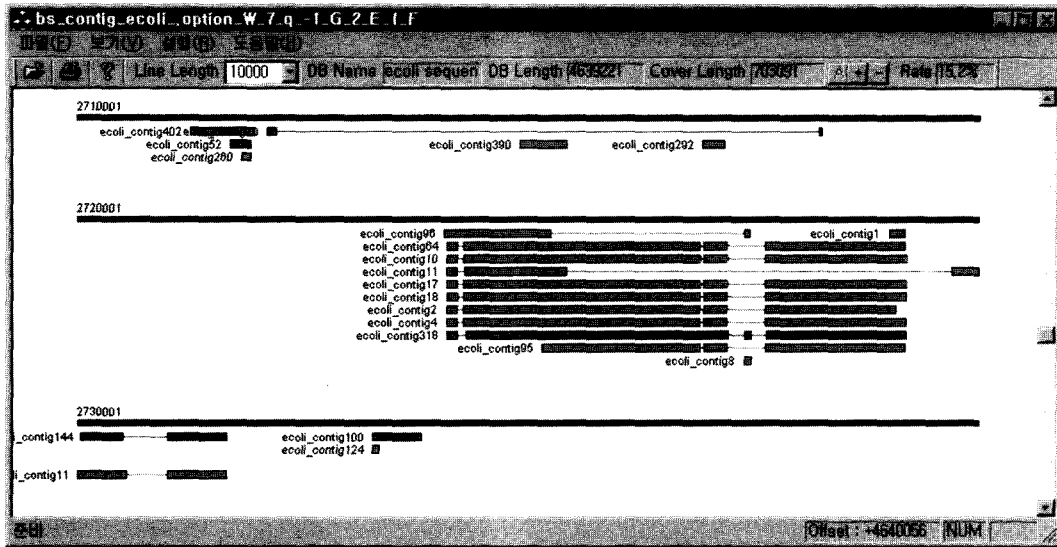


Fig. 3. Display window of a genome alignment. Black line, blue rectangles, and red rectangles mean DB sequence, covered regions aligned with forward sequences, and covered regions aligned with reverse sequences respectively. And blue/red lines mean uncovered regions in groups. The covered regions linked by linked lines belong to same group. The text preceding each group is the contig name containing the group.

결과 및 고찰

개발 프로그램의 기능과 사용 환경

Linux 상에서 실행된 BLAST나 FASTA 계산 파일의 결과와 대상이 되는 유전체 서열의 FASTA 파일을 입력으로 하여, 정렬 list 생성과 그래픽 표현을 위한 계산 프로그램을 Windows 상에서 실행한다.

입력할 파일들의 위치를 정해진 지시에 의해 선택하면, 이들을 모두 읽어 들여 대상 유전체에 대한 covered/uncovered 그래픽으로 출력(Fig. 3)한다.

윈도우의 윗부분에 그래픽 출력에 대한 보조 데이터가 표시되는데, scale을 표시하는 값이나, 전체 유전체 길이 및 covered 부

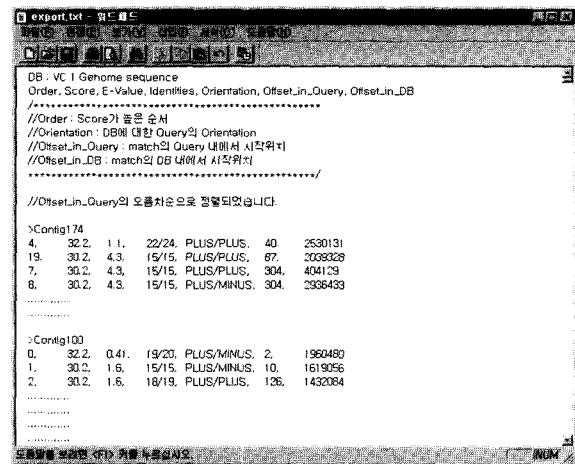


Fig. 5. The output file for alignment lists of all contigs. The text file of all alignment lists can be exported by the menu of alignment display window of Fig. 4. The sorting method can be selected. In this figure, the alignments are sorted by Offset\_in\_Query which are positions of alignments in the contig.

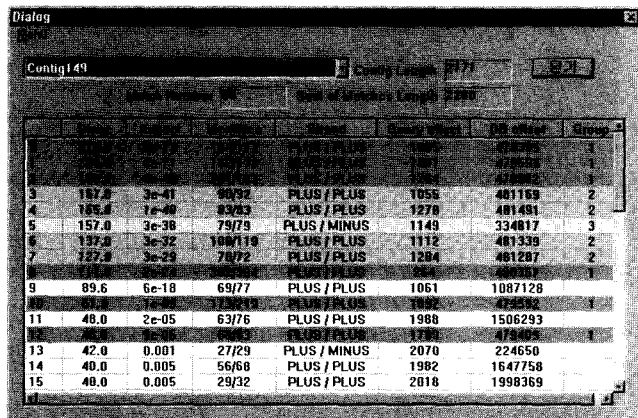


Fig. 4. Display window of alignment list of a contig. This dialog displays the list of all alignments of a chosen contig. The list is sorted ascendingly by E-value. The numbers of the last column are group numbers of the alignment. The white colored rows mean that the alignments are not included in any group.

분의 길이의 총합과 전체에 대한 covered의 길이 비율 등이 표시되고, scale 값을 변화시켜 그래픽 화면을 재출력할 수 있다.

한편, 각 contig에 대한 정렬 list의 자세한 내용을 보기 위해 contig 별 정렬 리스트 테이블(Fig. 4)을 출력할 수 있다. 이 출력 화면의 최상단에서 contig를 지정할 수 있으며, 화면 상단에 각 contig에 대한 정보로, 길이, 정렬의 개수, 정렬 길이의 합이 출력된다. 리스트에서는 각 정렬에 대한 정보로, 정렬 score, identity, 질의어 상에서의 위치와 대상 유전체 서열상에서의 위치 및 그룹 번호 등이 출력된다.

모든 contig들에 대한 정렬 정보를 통합하여 프린트 출력을 위해, 텍스트 파일(Fig. 5)을 생성할 수 있다.

**Table 2.** Result of GComp analysis for BH\_BS comparison pair

Cutoff E-value	10	1	0.1	0.01	0.001
number of effective alignments	6,864	6,335	5,419	4,901	4,567
highest E-value	4.1	0.45	0.05	0.006	6e-04
size of total cover region (bp)	1,913,524	1,904,254	1,883,114	1,867,176	1,856,073
cover ratio (%)	45.4	45.2	44.7	44.3	44.0

### 비교 유전체 시험 및 개별 사례 분석

개발된 프로그램의 시험과 실제의 미생물 유전체에 대한 비교 유전체 데이터 생성을 위해, *Bacillus halodurans* C-125(BH) (20), *Bacillus subtilis* (BS) (14), *Escherichia coli* K-12 MG1655 (K12) (6), *Escherichia coli* O157:H7 (O157) (12), *Vibrio cholerae* (VC) (13), 그리고 현재 아직 contig 단계에서 진행 중인 미생물의 유전체(CS)를 대상으로 계산을 수행하였다. 비교 쌍은 BH\_BS, K12\_O157, CS\_VC, BS\_K12로 하였고 cutoff E-value는 10을 default로 하였다.

BH\_BS 비교에서 BS sequence length인 4,214,814 bp 중 1,913,524 bp가 BH에 의해서 covered 영역으로 나타나서 cover ratio (전체 유전체 서열 길이에 대한 covered 서열의 길이 비율)는 45.4%로 나타났다. 높은 E-value를 가지는 정렬들이 cover ratio에 영향을 주는 정도를 알아보기 위해서 cuff off E-value를 다양하게 설정해 보았다. cutoff E-value를 10, 1.0, 0.1, 0.01, 0.001로 했을 때 각각의 cover ratio는 45.4%, 45.2%, 44.7%, 44.3%, 44.0%로 나타났다(Table 2).

이와 같은 조건으로 시험에 사용된 비교 쌍에 대해 같은 계산을 수행하였다(Table 3).

K12\_O157 비교 쌍은 79.5%의 cover ratio를 보여준다. Forward 정렬 그룹만의 cover ratio가 74.0%, reverse 정렬 그룹만의 cover ratio가 14.9%를 나타낸다.

CS\_VC 비교 쌍은 CS의 contig 서열들을 질의어로 입력했다. VC의 경우 염색체가 두 개이기 때문에 두 염색체의 서열을 VC1이 VC2로 따로 두어 계산하였다. VC1에서는 전체 2,961,149 bp에 대해 12.0%의 cover ratio를 나타내었고, VC2에서는 전체 1,072,315 bp에 대해 15.0%의 cover ratio를 나타내었다.

K12 유전체의 길이는 4,639,221 bp이며, BS 유전체의 길이는 4,214,814 bp이다. BS를 DB로 구축하고 K12를 질의어로 한 K12\_BS 비교 쌍의 경우 cover ratio는 19.2%로 나타났고, K12를 DB로 구축한 BS\_K12 비교 쌍의 경우는 17.1%로 나타났다.

각 경우에 나타난 유전체 상동성 결과에서, 핵산 수준의 유전체 서열에서는 일반적으로 진화적인 차이 정도로 예측되는 수준 이상으로 큰 차이가 나는 것을 볼 수 있었다.

### 상동성 분석 알고리즘에 대한 비교 유전체 분석 결과

FASTA를 사용했을 때의 비교 결과를 BLAST 실행 결과와 비교해 보기 위해, K12\_BS 비교 쌍을 대상으로 계산을 수행하였다. BS의 유전체 서열로 FASTA DB로 구축하고 K12의 유전체 서열을 20Kb 단위로 나누어 다수 개의 질의어로 입력한 FASTA

**Table 3.** GComp analysis for comparison pairs

Pair DB length (bp)	Query length (bp)	Cutoff E-value cover length (bp)	Cover ratio (%)
BH_BS	4,214,814	10	1,913,524 45.4
		0.01	1,867,176 44.3
K12_O157	5,528,970	10	4,397,839 79.5
		0.01	4,390,460 79.4
CS_VC1	2,961,149	10	354,338 12.0
		0.01	297,042 10.0
CS_VC2	1,072,315	10	161,079 15.0
		0.01	105,050 9.8
BS_K12	4,202,353	10	795,428 17.1
		0.01	700,617 15.1
K12_BS	4,639,221	10	810,089 19.2
		0.01	714,055 16.9

의 결과를 BLASTN 결과와 비교해 보았다. FASTA의 경우에는 질의어 길이에 제한이 있기 때문에 이러한 처리를 하였다. Display하기 위해서 그룹을 형성할 때 cutoff value를 0.01로 두었다. BLASTN의 결과는 19.2%로 나타났지만 FASTA의 결과는 12.7%로 나타났다. 이는 두 프로그램이 갖고 있는 알고리즘의 차이이기도 하지만, 파라미터를 동일 조건으로 둘 수 없는 프로그램 상의 차이이기도 하므로, BLAST의 실행 결과에 영향을 주는 파라미터를 다양하게 바꾸면 FASTA와 유사한 결과를 보이는 결과가 얻어질 수 있을 것으로 예측된다. 따라서 본 연구개발 결과를 포함하여 현재의 비교 유전체 분석 프로그램들은 파라미터에 대해 대단히 민감한 결과를 낼 수 있음을 알 수 있다. BLAST의 경우, 파라미터에 대한 결과의 차이에 대한 분석이 추후 연구에서 이루어질 것이며, 이를 기반으로 비교 유전체 분석 방법의 다양화나 표준화 및 결과 해석에 대한 기준 등에 대한 정보를 구할 수 있을 것이다.

핵산과 단백질 서열의 상동성의 결과 차이점을 고려해서 BS와 K12의 서열을 대상으로 TBLASTX로 상동성 계산을 처리하여 비교한 결과, BLASTN보다 다소 높은 cover ratio를 나타내었다. GenBank의 annotation된 K12 유전체 데이터에서 K12의 유전체 서열의 87.9%를 차지하는 coding region들을 아미노산 서열로 translation하고, BS를 대상으로 TBLASTN을 이용해서 상동성 계산을 처리하여 비교를 수행한 결과, 47.1%라는 높은 cover ratio

**Table 4.** K12\_BS comparison pair analysis with other BLAST programs

Data set	Program	Cover ratio (%)
Whole genome vs. whole genome	BLASTN	19.2
	TBLASTX	21.8
Coding region set (proteins) vs. whole genome	TBLASTX	30.0
Coding region set (nucleotides) vs. coding region set (proteins)	TBLASTN	47.1

를 보였다.

TBLASTX나 TBLASTN으로 전처리한 결과의 계산을 통해, 핵산 수준의 비교와 단백질 수준의 비교가 큰 차이를 보이고 있음(Table 4)을 볼 수 있는데, coding region과 non-coding region의 차이뿐만 아니라, coding region에 대해서도 핵산 수준의 비교와 단백질 수준에서의 비교가 큰 차이를 보이고 있음을 나타내고 있다.

한편, K12의 coding region들을 K12 유전체와 비교의 경우에도 앞의 분석과 다소 다른 결과가 나타나는 것을 볼 수 있었다(Table 4). 상동성 있는 유전자들의 분포를 화면으로 표시할 수 있었고, 또한 repeat나 multiple copy의 유전자 분포도 볼 수 있었다. 많은 부분에서 여러 유전자들이 겹쳐서 나타났는데, 이 중 세 곳 이상에서 높은 homology를 보이는 유전자들도 다수 나타났다. 그 예로, IS1 protein InsB의 경우 gene size 504 bp인데, 6곳에서 E-value가 0.0으로 나타났고, IS5 transposase의 경우 1,017 bp이며 무려 11 곳에서 E-value가 0.0으로 나타났다. 이처럼 한 유전체의 자체 상동성 분석에 응용하기 위해 이 프로그램을 유용하게 사용할 수 있음을 볼 수 있었다.

**개발 환경 및 수행 속도**

GComp는 연구자들이 사용법에 어려움을 느끼지 않게 하기 위해 윈도우즈 환경에서, 윈도우즈 프로그래밍을 하기에 가장 적합한 개발 도구인 VC++에 의해서 개발되었다. VC++는 객체 지향 프로그래밍인 C++에 기반을 두었기 때문에 각 module를 추가하거나 고치기에 적당하고, 객체 지향이라는 개념과 함께 포인터를 사용하기 때문에 데이터의 처리속도가 빠르다.

Intel Pentium III 750의 CPU와 256 Mbyte의 메모리를 장착한 Windows98/SE를 수행 환경으로 해서 유전체 비교 쌍에 대해서 GComp의 수행 속도를 측정하였다(Table 5). 유사성이 매우 높은 K12\_O157의 경우를 제외하고는 수행속도가 거의 유사하게 나온

**Table 5.** GComp running time analysis

Pair (BLASTN)	Running time (sec)
BH_BS	5.60
K12_O157	31.89
BS_K12	5.22
K12_BS	5.56

것을 알 수 있다. 이는 유사성이 높을수록 낮은 E-value를 가지는 정렬의 개수가 많아지기 때문에 이를 재조합하는 과정에서 계산의 양이 많아지기 때문이다.

**기능 및 알고리즘 개선 방향**

본 연구의 개발 결과 분석과 유전체 사례 분석 연구를 통해, 초기에 설계했던 프로그램의 기능 외에 인터페이스와 분석의 종류 면에서 추후 연구를 통해 개발할 부분의 필요성과 방향에 대해 설계를 할 수 있었다. 비교 유전체를 위한 도구의 요구사항이 현재 뚜렷하지 않고 개발의 여지가 많아, 유전체 연구에 대한 심도 있는 분석을 통해 이러한 도구의 사양과 이를 해결하는 알고리즘에 대한 개발할 필요가 많을 것으로 생각된다.

BLAST의 파라미터에 따른 분석결과와 차이점에 대한 추후 분석은, 비교 쌍의 특성에 대해 파라미터를 설정하는 방법과 일반적인 비교에서의 파라미터 표준화 내지 최적화에 대한 정보를 제공해 줄 것이다. Coding region과 non-coding region에 대한 별도의 분석과 구역별 상동성의 특징 분석을 통해, 유전체 비교 방법에 대한 일반적인 프로토콜과 표준 데이터에 대한 정보를 구할 수 있을 것이다. 한편, 이를 위해 gene prediction 프로그램과의 연계 및 통합을 통해 보다 효과적이고 통합적인 도구를 개발할 수 있을 것이다.

BLAST나 FASTA를 효과적으로 사용하기 위해 1차적으로 Unix 서버에서 계산을 수행한 후에 Windows용 GComp를 사용해야 하므로(Windows 버전의 BLAST는 여전히 계산 용량 등의 면에서 제약이 많으므로), 대부분의 생물학 연구자에게는 사용 환경상 큰 제약이 생기게 된다. 따라서 Windows 인터페이스를 통해 Unix 상의 BLAST와 FASTA를 실행할 수 있는 기능을 개발하는 것이 주요한 사항일 것이다. 한편, 대상 유전체의 길이에 제한이 있는 FASTA의 제한점을 간접적으로 보완하여 실질적으로 FASTA를 비교유전체 분석에 활용하기 위해, 유전체 서열을 다수의 서열로 자동 분할하여 계산하고 그 결과를 자동 통합하여, 비교 유전체 분석 결과에 반영하는 기능을 구현할 필요가 있을 것이다.

본 프로그램은 유전체 비교 분석 뿐 아니라 하나의 유전체에 대한 분석을 위해서도 활용될 수 있다. 즉, repeat region이나, gene의 분포나 gene의 cluster 분포 등을 위해서 활용될 수 있을 것이므로, 이를 직접적으로 처리할 수 있는 기능을 구현하면, 보다 효과적이고 활용도가 높은 비교유전체 분석도구로 활용될 수 있을 것이다. 한편, 미생물 유전체 프로젝트 수행용 도구와 통합하여, 미생물 프로젝트의 수행 도중에 이 도구를 보다 효율적으로 활용할 수 있도록 할 수 있을 것이다.

GComp는 <http://www.smallsoft.co.kr/~hstae/GComp/GComp.html>에서 다운로드 받을 수 있다.

**참고문헌**

1. Altschul, S.F., W. Gish, W. Miller, E.W. Myers, and D.J. Lipman, 1990. Basic local alignment search tool. *J. Mol. Biol.* 215, 403-410.
2. Altschul, S.F., T.L. Madden, A.A. Schaffer, J. Zhang, Z. Zhang, W.

- Miller, and D.J. Lipman, 1997. Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic Acids Research* 25, 3389-3402.
3. Bansal, A.K., P. Bork, and P.J. Stuckey, 1998. Automated pairwise comparisons of microbial genomes. *Math. Modelling and Sci. Computing* 9, 1-23.
  4. Bansal, A.K., 1999. An automated comparative analysis of 17 complete microbial genomes. *Bioinformatics* 15, 900-908.
  5. Bateman, A., E. Birney, L. Cerruti, R. Durbin, L. Etwiller, S.R. Eddy, S.G. Jones, K.L. Howe, M. Marshall, and E.L.L. Sonnhammer. 2002 The Pfam Protein Families Database. *Nucleic Acids Res.* 30, 276-280.
  6. Blattner, F.R., G. Plunkett, C.A. Bloch, N.T. Perna, V. Burland, M. Riley, J. Collado-Vides, J.D. Glasner, C.K. Rode, G.F. Mayhew, J. Gregor, N.W. Davis, H.A. Kirkpatrick, M.A. Goeden, D.J. Rose, B. Mau, Y. Shao, 1997. The Complete Genome Sequence of *Escherichia coli* K-12. *Science* 277, 1453-1462.
  7. Delcher, A.L., S Kasif, R.D. Fleischmann, J. Peterson, O. White, and S.L. Salzberg, 1999. Alignment of whole genomes. *Nucleic Acids Res.* 27, 2369-2376.
  8. Delcher, A.L., A. Phillippy, J. Carlton, and S.L. Salzberg, 2002. Fast algorithms for large-scale genome alignment and comparison. *Nucleic Acids Res.* 30, 2478-2483.
  9. Fleischmann, R.D., M.D. Adams, O. White, R.A. Clayton, E.F. Kirkness, A.R. Kerlavage, C.J. Bult, J.F. Tomb, B.A. Dougherty, and J.M. Merrick, 1995. Whole-genome random sequencing and assembly of *Haemophilus influenzae* Rd. *Science* 269, 496-512.
  10. Folrea, L., C. Riemer, S. Schwartz, Z. Zhang, N. Stojanovic, W. Miller, and M. McClelland, 2000. Web-based visualization tools for bacterial genome alignments. *Nucleic Acids Res.* 20, 3486-3496.
  11. Gusfield, D., 1997. Algorithms on Strings, Trees, and Sequences. Computer Science and Computational Biology. Cambridge University Press, New York.
  12. Hayashi, T., K. Makino, M. Ohnishi, K. Kurokawa, K. Ishii, K. Yokoyama, C.G. Han, E. Ohtsubo, K. Nakayama, T. Murata, M. Tanaka, T. Tobe, T. Iida, G. Takami, T. Honda, C. Sasakawa, N. Ogasawara, T. Yasunaga, S. Kuhara, T. Shiba, M. Hattori, and H. Shinagawa. 2001. Complete genome sequence of enterohemorrhagic *Escherichia coli* O157:H7 and genomic comparison with a laboratory strain K-12. *DNA Res.* 8, 11-22.
  13. Heidelberg, J.F., J.A. Eisen, W.C. Nelson, R.A. Clayton, M.L. Gwinn, R.J. Dodson, D.H. Haft, E.K. Hickey, J.D. Peterson, L. Umayam, S.R. Gill, K.E. Nelson, T.D. Read, H. Tettelin, D. Richardson, M.D. Ermolaeva, J. Vamathevan, S. Bass, H. Qin, I. Dragoi, P. Sellers, L. McDonald, T. Utterback, R.D. Fleischmann, W. C. Nierman, O. White, S.L. Salzberg, H.O. Smith, R.R. Colwell, J.J. Mekalanos, J.C. Venter, and C.M. Fraser. 2000. DNA sequence of both chromosomes of the cholera pathogen *Vibrio cholerae*. *Nature* 406, 477-483.
  14. Kunst, F., N. Ogasawara, I. Moszer, A.M. Albertini, G. Alloni, V. Azevedo, M.G. Bertero, P. Bessieres, A. Bolotin, S. Borchert, R. Borriss, L. Boursier, A. Brans, M. Braun, S.C. Brignell, S. Bron, S. Brouillet, C.V. Bruschi, B. Caldwell, V. Capuano, N.M. Carter, S.K. Choi, J.J. Codani, I.F. Connerton, N.J. Cummings, R.A. Daniel, F. Denizot, K.M. Devine, A. Dsuterhoft, S.D. Ehrlich, P.T. Emmerson, K.D. Entian, J. Errington, C. Fabret, E. Ferrari, D. Foulger, C. Fritz, M. Fujita, Y. Fujita, S. Fuma, A. Galizzi, N. Galeron, S.Y. Ghim, P. Glaser, A. Goffeau, E. J. Golightly, G. Grandi, G. Guiseppe, B.J. Guy, K. Haga, J. Haech, C.R. Harwood, A. Henaut, H. Hilbert, S. Holsappel, S. Hosono, M.F. Hullo, M. Itaya, L. Jones, B. Joris, D. Karamata, Y. Kasahara, M. Klaerr-Blanchard, C. Klein, Y. Kobayashi, P. Koetter, G. Koningstein, S. Krogh, M. Kumano, K. Kurita, A. Lapidus, S. Lardinois, J. Lauber, V. Lazarevic, S.M. Lee, A. Levine, H. Liu, S. Masuda, C. Mauel, C. Medigue, N. Medina, R.P. Mellado, M. Mizuno, D. Moestl, S. Nakai, M. Noback, D. Noone, M. O'Reilly, K. Ogawa, A. Ogiwara, B. Oudega, S.H. Park, V. Parro, T.M. Pohl, D. Portetelle, S. Porwollik, A.M. Prescott, E. Presecan, P. Pujic, B. Purnelle, G. Rapoport, M. Rey, S. Reynolds, M. Rieger, C. Rivolta, E. Rocha, B. Roche, M. Rose, Y. Sadaie, T. Sato, E. Scanlan, S. Schleich, R. Schroeter, F. Scoffone, J. Sekiguchi, A. Sekowska, S.J. Seror, P. Serror, B.S. Shin, B. Soldo, A. Sorokin, E. Tacconi, T. Takagi, H. Takahashi, K. Takemaru, M. Takeuchi, A. Tamakoshi, T. Tanaka, P. Terpstra, A. Tognoni, V.Tosato, S. Uchiyama, M. Vandenbol, F. Vannier, A. Vassarotti, A. Viari, R.Wambutt, E.Wedler, H.Wedler, T.Weitzenegger, P. Winters, A. Wipat, H. Yamamoto, K. Yamane, K. Yasumoto, K. Yata, K. Yoshida, H.F. Yoshikawa, E. Zumstein, H. Yoshikawa, and A. Danchin. 1997. The complete genome sequence of the gram-positive bacterium *Bacillus subtilis*. *Nature* 390, 237-238.
  15. Mayor, C., M. Brudno, J.R. Schwartz, A. Poliakov, E.M. Rubin, K.A. Frazer, L.S. Pachter, and I. Dubchak. 2000. VISTA : visualizing global DNA sequence alignments of arbitrary length. *Bioinformatics* 16, 1046-1047.
  16. Pearson, W.R., 1990. Rapid and Sensitive Sequence Comparison with FASTP and FASTA. *Methods Enzymol.* 183, 63-98.
  17. Pearson, W.R., 2000. Flexible similarity searching with the FASTA3 program package. *Methods Mol. Biol.* 132, 185-219.
  18. Roten, C.H., P. Gamba, J. Barblan, and D Karamata. 2002. Comparative Genomics (CG): a database dedicated to biometric comparisons of whole genomes. *Nucleic Acids Res.* 30, 142-144.
  19. Schwartz, S., Z. Zhang, K.A. Frazer, A. Smit, C. Riemer, J. Bouck, R. Gibbs, R. Hardison, and W Miller. 2000. PipMaker. A Web Server for Aligning Two Genomic DNA Sequences. *Genome Res.* 10, 577-586.
  20. Takami, H., K. Nakasone, Y. Takaki, G. Maeno, R. Sasaki, N. Masui, F. Fuji, C. Hirama, Y. Nakamura, N. Ogasawara, S. Kuhara, and K. Horikoshi. 2000 Complete genome sequence of the alkaliphilic bacterium *Bacillus halodurans* and genomic sequence comparison with *Bacillus subtilis*. *Nucleic Acids Res.* 28, 4317-4331.
  21. Tatusov, R.L., M.Y. Galperin, D.A. Natale, and E.V. Koonin. 2001 The COG database: a tool for genome-scale analysis of protein functions and evolution. *Nucleic Acids Res.* 29, 22-28.
  22. Walker, M., V. Pavlovic and S. Kasif. 2002 A comparative genomic method for computational identification of prokaryotic translation initiation sites. *Nucleic Acids Res.* 30, 3181-3191.
  23. Xie, H., A. Wasserman, Z. Levine, A. Novik, V. Grebinskiy, A. Shoshan, and L. Mintz. 2002. Large-Scale Protein Annotation through Gene Ontology. *Genome Res.* 12, 785-794.
  24. Zdobnov, E.M., C. Mering, I. Letunic, D. Torrents, M. Suyama, R.R. Copley, G.K. Christophides, D. Thomasova, R.A. Holt, G.M. Subramanian, H.M. Mueller, G. Dimopoulos, J.H. Law, M.A. Wells, E. Birney, R. Charlab, A.L. Halpern, E. Kokoza, C.L. Kraft, Z. Lai, S. Lewis, C. Louis, C. Barillas-Mury, D. Nusskern, G.M. Rubin, S.L. Salzberg, G.G. Sutton, P. Topalis, R. Wides, P. Wincker, M. Yandell, F.H. Collins, J. Ribeiro, W.M. Gelbart, F.C. Kafatos, and P. Bork. 2002. Comparative Genome and Proteome Analysis of *Anopheles gambiae* and *Drosophila melanogaster*. *Science* 298, 149-159.

(Received October 29, 2002/Accepted November 21, 2002)



---

**ABSTRACT: A Genomics Tool for Microbial Genome Comparison Using BLAST/FASTA**

**Hongseok Tae<sup>1</sup>, Daesang Lee<sup>2</sup>, Wan Park<sup>1</sup>, and Kiejung Park<sup>2,\*</sup>** (<sup>1</sup>Department of Microbiology, Kyungpook National University, Daegu 762-701, Korea, <sup>2</sup>Information and Technology Institute, SmallSoft Co., Ltd., Daejeon 305-811, Korea)

We have developed GComp as an analysis tool for microbial genome comparison. This tool exploits BLAST or FASTA as a preprocessing program for local alignments to detect homologous regions, parses the homology search results, and generates tables and files to show homology relationship between two genomes at a glance. The interface for graphical representation of the comparative genomic analysis has been also implemented. Our test cases shows that the program can be useful in practice for intuitive and quantitative comparison of microbial genome sequence pairs as well as self-genome analysis. A few additional features have been devised and designed, which will be added in the further development.