The Korean Journal of Microbiology, Vol. 35, No. 2, June 1999, p. 121-127

염기서열 해독작업을 위한 핵산 단편 조립 프로그램의 개발

이 동 훈*
충북대학교 자연과학대학 생명과학부

염기서열 해독작업에서 각 핵산 단편을 조립하는 contig 구성문제에 활용이 가능한 computer program (SeqEditor)을 개발하였다. 본 프로그램은 국내에서 광범위하게 사용되고 있는 MS-Windows 운영체제의 개인용 컴퓨터에서 작동이 가능하며, GenBank, FASTA, ASCII 등과 같은 다양한 형태의 염기서열 자료를 입력할 수 있다. 두 단편에서 최대 유사도를 나타내는 부분을 정렬하는 작업에는 염기서열의 국부적 상동성(local homology)을 계산하고 dynamic programming 알고리즘을 적용하는 방법을 이용하였다. 또한 사용하기 편리한 그래픽 방식의 인터페이스를 제공하여 초보자도 손쉽게 조작할 수 있다는 장점을 갖는다. 본 프로그램의 성능을 검증하기 위하여 세균과 곰팡이로부터 해독된 16S rRNA와 18S rRNA 유전자의 단편 염기서열을 재구성하는 작업에 프로그램을 사용하였을 때에 효율적인 작업이 가능하였다.

KEY WORDS □ contig assembly, DNA sequencing, dynamic programming algorithm, SeqEditor, windows program

대부분의 분자생물학적 연구에서, 특정 유전자(gene)의 염기서열을 해독(sequencing)하고 이미 보고된 유사한 자료들과 비교, 분석하는 작업은 자주 이용되는 연구 방법이다. 최근에는 자동화된 염기서열 분석장치(automatic sequence analyzer)를 이용하여 염기서열 해독작업에 요구되는 인적 노력과 소요 시간을 줄일 수 있게 됨에 따라 많은 연구가 활발히 이뤄지고 있다. 그러나 현재까지 염기서열의 길이가 매우 긴 특정 유전자의 전체 염기서열을 한 번의 실험만으로 모두 얻을 수 있는 방법은 없다. 전체 염기서열을 얻기 위해서는 먼저 특정 유전자를 여러 개의 단편(fragment)으로 구분한 후에 각 단편들의 염기서열을 분석하여야 한다. 그리고 염기서열이 밝혀진 각 단편들의 정보를 이용하여 전체 염기서열을 재구성하여야 하며, 이러한 작업을 contig 구성 작업이라 한다(20). 독립된 실험에서 확인될 수 있는 각 단편들의 정보량은 일반적으로 수백 염기쌍(nucleotide base pair)을 넘지 못한다. 따라서 수천 내지 수만 개의 염기로 이루어진 긴 염기서열을 얻기 위해서는 수십 또는 수백 개 이상의 단편들의 정보를 재구성하여야 한다. 각 단편들의 염기서열을 재구성하는 작업은 단순하며 반복적이지만 상당히 많은 염기의 비교가 요구된다. 즉 각 단편들의 5'-, 3'-말단의 염기서열을 비교하여 일치하는 염기서열의 길이가 가장 긴 구성이 되도록 각 단편들을 배치하여야 한다. 현재까지 단편 염기서열의 재구성 작업은 주로 실험자의 직관력에 의존하는 작업을 통하여 수행되고 있다. 그러나 각 단편들의 염기서열을 해독할 때에 말단부위에서는 실험오차가 증가하기 때문에 잘못된 염기서열을 얻는 경향이 있으며, 오류가 있는 정보는 단편의 재구성 작업을 어렵게 한다. 특히 염기서열 분석장치를 이용할 때, 동시에 다수의 시료를 분석할 수 있어 짧은 시간 내에 많은 결과를 얻을 수 있으나 각 단편들의 말단에서 더욱 심각한 오류가 발견되는 경향이 있다. 따라서 염기서열 분석장치의 결과들을 재구성하는 작업은 작업량 뿐만이 아니라 작업의 난이도가 함께 증가하며 많은 작업시간을 요구한다.

또한 분자생물학적 연구가 진행됨에 따라 축적되는 정보량이 증가하게 되며 실험결과의 분석이 점차로 복잡해지고 있다. 따라서 컴퓨터를 이용한 분자생물학적 정보의 처리 및 분석이 시도되었으며, 현재까지 다양한 기종의 컴퓨터에서 작동되는 많은 프로그램들이 개발되었다. 국내에서도 플라스미드의 제한효소 지도 작성을 위한 프로그램(13)이 개발된 이후로 많은 프로그램들이 분자생물학적 자료를 처리하기 위하여 개발되었다. 최근에는 ribosomal RNA 염기서열의 다중정렬(multialignment) 및 계통관계 분석을 위한 프로그램(1)과 컴퓨터 모델링을 이용하여 이차구조를 예측하기 위한 프로그램(4)도 보고된 바 있다. 핵산의 염기서열은 A, G, C, T의 4개의 문자로 구성된 단순한 구조의 문자열로 나타낼 수 있으며, 이러한 염기 문자열의 비교 연산에는 컴퓨터의 빠른 계산능력을 이용하면 작업의 효율성을 높일 수 있다. 그러므로 많은 염기를 비교하여 각 단편들의 염기서열을 재구성하는 작업을 수행할 수 있는 프로그램의 개발에도 많은 연구가 진행되었다. 각 단편의 염기서열 정보를 이용하여 전체염기서열을 재구성 할 수 있는 프로그램으로는 SEQAID(Seqencing Aid)(17), CAP (Contig Assembly Program)(10), CAP2(Contig Assembly Program 2)(9), GAP(Genome Assembly Program)(6) 등이 작성되었으며, 국내에서도 FAP(Fragment Assembly Program)(3)과 XFAP(X-window-based Fragment Assembly Program)(2)이 개발되었다. 그러나 위에서 언급된 프로그램들은 모두 유닉스(UNIX)를 운영체제(operating system)로 하는 워크스테이션(workstation) 또는 그 이상의 장비(hardware)를 필요로 한다. 또한 대부분의 프로그램이 제공하는 사용자 환경이 문자 위주의 텍스트 방식이기 때문에 실험자가 사용하기에 불편한 점이 많다. 따라서 본 연구에서는 컴퓨터의 사용에 익숙하지 않은 분자생물학 연구자도 염기서열 해독작업 과정에서 손쉽

*To whom correspondence should be addressed
Tel : 0431-261-3261, Fax : 0431-264-9600
E-mail : donghun@cbucc.chungbuk.ac.kr

게 이용할 수 있는 단편 염기서열 재구성 프로그램을 개발하고자 하였다. 이미 보고된 프로그램들이 다양한 분석기능을 제공하지만 고가의 장비들을 요구하는 반면에, 본 연구에서는 contig 구성 작업에 실질적으로 도움을 줄 수 있는 단순한 기능만을 제공하지만 실험실에서 흔히 이용되는 저가의 개인용 컴퓨터에서도 사용할 수 있으며 그래픽 사용자 환경(graphical user interface)을 제공하는데 중점을 두었다.

재료 및 방법

프로그램 개발환경

본 프로그램은 IBM PC와 호환되는 개인용 컴퓨터에서 작동이 가능하다. 초보자도 간편하게 프로그램을 이용할 수 있도록 그래픽 사용자환경을 추가하는 작업은 Microsoft Windows 환경에서 수행되는 형식을 이용하여 프로그램 개발 작업의 효율성을 높였다. *Windows 95/98*과 *NT*를 운영체제로 이용할 경우에는 운영체제 자체가 그래픽 사용자 환경을 제공하고 있으며, MS-DOS를 운영체제로 이용할 경우에는 Windows 3.1 프로그램을 실행시킴으로써 그래픽 사용자 환경을 얻을 수 있다. Windows 환경에서 사용할 수 있는 프로그램 개발언어는 여러 종류가 있지만 본 연구에서는 Windows용 프로그램을 빠른 시간 내에 효율적으로 개발할 수 있는 Microsoft사의 Visual Basic 언어를 이용하였다. 또한 Windows 계열(*Windows 3.1/95/98/NT*)의 운영체제에 맞는 두 종류의 실행파일을 작성하였다. 즉 멀티프로세싱 등과 같은 Windows 95/98/NT의 장점을 충분히 활용할 수 있는 32 비트 실행파일을 작성함과 동시에 Windows 3.1 환경에서 필요한 16 비트 실행파일도 함께 작성하여 성능이 떨어지는 구형의 기종도 활용할 수 있도록 하였다.

단편정렬 방법

본 프로그램의 주목적인 단편의 정렬 작업에는 국부적 상동성(local homology)을 구하는 문제가 해결되어야 하며 Smith-Waterman(19)의 알고리즘(algorithm)을 개선한 Huang(10)의 방법을 변형하였다. 두 단편의 염기서열이 $A=a_1a_2a_3\cdots\cdots a_N$와 $B=b_1b_2b_3\cdots\cdots b_M$이고, 두 단편의 i번째와 j번째 염기사이의 유사도는 $S(a_i, b_j)$로 나타내며, q는 염기가 결실(deletion) 또는 삽입(insertion)되어 공백(gap)을 형성하였을 경우 벌점(penalty)을 주기 위한 값이라고 할 때, 유사도 값이 최대가 되는 부분(*segment*)의 정렬을 찾을 수 있는 수식은 아래와 같은 행렬 H로 나타낼 수 있으며 수식 (3)에 의해 최대값을 갖는 원소의 위치를 확인한다. 그후 이 원소가 가장 큰 값을 갖도록 기여한 원소들을 순서대로 역추적하면 최대값 정렬을 갖는 부분(segment)의 정보를 알 수 있다.

$$H_{i,j}=0,\ \text{if } i=0 \text{ or } j=0 \quad (1)$$

$$H_{i,j}=\text{Max}\{H_{i-1,j-1}+s(a_i, b_j),\ H_{i-1,j}-q,\ H_{i,j-1}-q\},\ \text{if } i>0 \text{ and } i>0 \quad (2)$$

$$\text{Maximal value}=\text{Max}\{H_{i,M}, H_{N,j};\ 1\le i\le N \text{ and } 1\le j\le M\} \quad (3)$$

본 연구에서도 빠르고 효율적인 행렬계산을 위하여 수식 (1)과 (2)를 이용하였으며, 추적방향 행렬도 함께 계산하는 dynamic programing 방법(21)을 사용하였다. 그러나 단편을 정렬하기 위한 역추적의 출발점을 단순히 수식 (3)의 방법에 의해 결정하지 않았으며, 행렬값 계산시에 항상 최대값을 갖는 원소 위치를 기억하여 필요한 경우 추적방향 행렬 변환에 사용하였다. 최대 값을 갖는 원소가 행렬의 마지막 행 또는 열에 위치하지 않을 경우, 최대값을 갖는 원소의 위치를 기준으로 추적방향 행렬을 수정하였으며 수식 (4)를 이용하여 행렬의 마지막 행 또는 열중에서 최대값을 갖는 새로운 원소를 재선정하였다.

$$\text{New value}=\text{Max}\{H_{i,M}, H_{N,j};\ k\le i\le N \text{ and } 1\le j\le M\},\ \text{if } H_{k,l}=\text{Maximal value} \quad (4)$$

따라서 본 연구에서 사용된 방법은 말단 부위의 오류로 인하여 최대값을 갖는 원소가 행렬의 끝부분에 위치하지 않더라도 역추적시 행렬의 중간에 위치하는 최대값을 갖는 원소를 반드시 통과할 수 있도록 개선되었다.

염기서열 분석

프로그램의 성능을 검증하기 위하여 세균의 16S rRNA와 곰팡이의 18S rRNA 염기서열 분석작업에 활용하였다. 세균 군집의 시료는 Lee 등(12)의 방법으로 자연환경으로부터 직접 핵산을 추출하였으며, 곰팡이 시료는 *Penicillium*속 균주를 PDB배지에서 진탕배양(24C, 48 hr)한 후에 나일론 필터로 *mycelium*을 수확하고, *Marmur* 방법(*15*)을 이용하여 핵산을 추출하였다. 핵산시료로부터 16S rRNA와 18S rRNA 유전자를 선택적으로 증폭하기 위하여 universal primer set인 8F, 1492R, NS1, NS8을 이용하여 programable thermal controller(MJ Research Inc., PTC-100)로 PCR을 수행하였다(11, 22). 증폭된 핵산은 nucleic acid purification kit (Bioneer Inc., DNA PrepMate)로 정제한 후에 ABI PRISM™automatic sequencer(Perkin Elmer)를 사용하여 direct sequencing을 수행하였다(*14, 18*). 하나의 *primer*당 *300-500 bp*의 염기서열이 분석되었고, 각 단편의 염기서열 정보와 본 연구에서 개발된 프로그램을 이용하여 16S rRNA와 18S rRNA 유전자의 전체 염기서열을 재구성하였다.

결과 및 고찰

프로그램 동작환경

현재 국내에서 광범위하게 사용되는 개인용 컴퓨터로는 IBM PC와 호환성을 갖는 제품이 주류를 이루고 있으며 대부분의 사용자가 MS-DOS 또는 Windows 95/98을 운영체제로 이용하고 있다. UNIX를 운영체제로 하는 중형 컴퓨터는 보다 안정적인 성능을 나타낼 수 있으나 고가의 장비를 요구하며 초보자가 컴퓨터를 작동하기 어렵다는 단점이 있다. 반면에 개인용 컴퓨터는 최근에 성능 향상이 매우 빠른 속도로 이루어지고 있으며 가격도 비교적 저렴하다. 따라서 본 연구에서 개발되는 프로그램이 요구하는 컴퓨터의 성능이 개인용 컴퓨터로도 충분히 충족될 수 있다고 판단되어 PC 기종에서 작동이 가능한 프로그램을 개발하였다. 또한 본 프로그램은 32 비트와 16 비트 두 종류의 실행파일로 번역하기 위하여 개발 단

계에서 하위 호환성을 갖도록 작성되었다. 현재까지 사용되고 있는 구형의 컴퓨터는 성능의 제약 때문에 Windows 95를 운영체제로 사용할 수 없으며 MS-DOS를 운영체제로 사용하고 있다. MS-DOS를 운영체제로 이용하는 컴퓨터에서는 Windows 3.1 프로그램을 설치하면 16 비트 파일로 번역된 프로그램을 사용할 수 있다. 따라서 두 종류의 실행파일로 구성된 본 프로그램은 국내에서 광범위하게 사용되고 있는 MS-DOS 및 Windows 계열(Windows 95/98/NT)의 운영체제를 갖는 저가의 개인용 컴퓨터에서 동작이 가능하며 구형의 컴퓨터도 이용함으로써 실험실의 자원을 효율적으로 이용할 수 있게 한다.

사용자 환경 및 사용자 인터페이스

대부분의 단편조립 프로그램이 제공하는 사용자 환경이 문자위주의 텍스트 방식이기 때문에 실험자가 사용하기에 불편한 점이 많다. 그러므로 본 연구에서는 컴퓨터의 사용에 익숙하지 못한 분자생물학 연구자도 염기서열 해독작업 과정에서 손쉽게 이용할 수 있는 단편 염기서열 재구성 프로그램을 개발하고자 하였다. 그래픽 처리방식의 사용자 인터페이스는 사용자가 쉽게 시스템을 이해하고 사용할 수 있을 뿐 아니라 자료처리를 쉽게 여러 형태로 할 수 있어 시스템 수행의 효율을 높일 수 있다. 특히 윈도우를 이용하는 프로그램은 화면 전체가 아닌 사용자가 지정한 영역을 하나의 화면처럼 사용할 수 있게 해주어 여러 화면을 다루는 효과를 보일 수 있으며 한 화면상에 부메뉴 처리 등을 할 수 있다. 이 경우 한 화면보다 큰 자료의 표현에는 스크롤 기능을 이용하기도 한다. 본 프로그램에서도 그래픽 방식의 사용자 인터페이스를 채택하였으며 입, 출력에서 사용자의 용이한 조작에 의한 즉각적인 편집이나 화면출력 및 프린터출력 등의 기능을 가질 수 있도록 하였다. 따라서 초보자도 손쉽게 조작할 수 있다는 장점을 갖으며, 동시에 여러 개의 단편조립 윈도우를 만들 수 있으므로 작업효율을 높일 수 있다.

프로그램의 주요 기능 및 구조

현재까지 보고된 단편정렬 프로그램들은 여러 개의 단편들을 동시에 처리하는 기능이 있으나 이를 처리하기 위한 빠른 계산능력과 많은 기억장치를 필요로 하기 때문에 중형 컴퓨터 이상의 장비가 필요하다. 일반적으로 단편조립 프로그램의 실행속도를 제한하는 것은 대부분 각 단편들로 구성된 쌍으로부터 최대값 정렬을 구하기 위한 dynamic programming 알고리즘을 적용하는 단계 때문이다. 또한 2차원 배열을 데이터 구조로 사용하는 dynamic programming 알고리즘에서는 입력자료의 크기가 큰 경우에 메모리 부족 현상을 야기할 수도 있다 따라서 느린 실행 속도와 메모리의 부족현상 때문에 개인용 컴퓨터에서 운영이 가능한 프로그램의 개발이 어렵다. 그러나 실제로 염기서열을 해독하는 과정에서 shotgun sequencing과 같은 소수의 실험을 제외하고는 동시에 여러 개의 단편을 조립할 필요가 없다. 최근의 염기서열 해독작업에는 PCR 및 direct sequencing 방법들이 자주 사용되며, 실험자가 적절한 primer를 설계함에 따라 각 단편들의 정보를 순서대로 배치할 수 있다. 실험자가 각 단편들의 순서를 파악할 수 없는 경우에는 여러 개의 단편들을 동시에 처리하여 contig를 구성

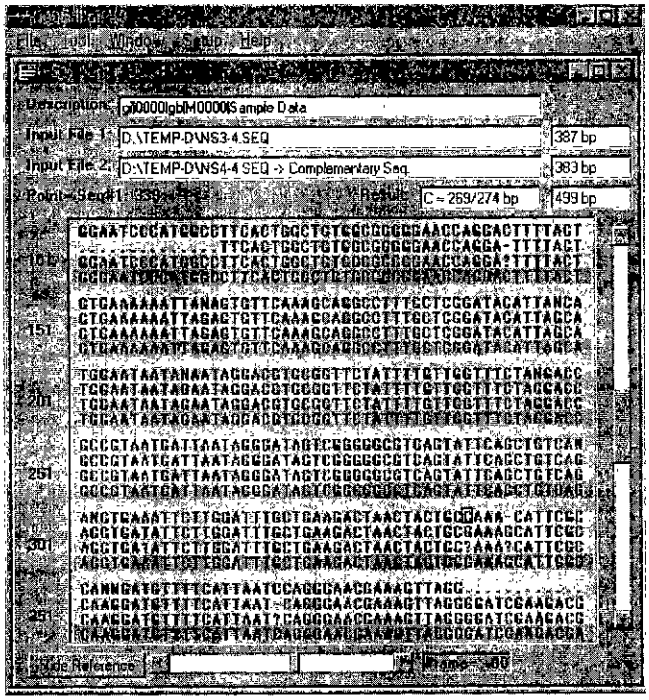

Fig. 1. The example of assembing short DNA fragments with SeqEditor. The 18S rDNA sequences of *P. purprogenome* are constructed from the fragments sequenced by a automatic sequencer. The consensus sequences are compared with the reference sequences of *P. marneffei* (GenBank Accession No : AF034197).

할 필요가 있으나, 각 단편들의 순서가 명확한 경우에는 한 번에 2개의 단편만을 처리하여도 충분하다. 실험자가 인지하고 있는 순서대로 각 단편들을 추가 조립하면 특정 유전자의 전체 염기서열을 얻을 수 있으며, 프로그램은 두 단편의 염기서열이 최대 유사도를 갖도록 중첩시키는 역할만을 수행하면 된다. 본 연구에서는 한 번에 2개의 단편만을 처리하도록 함으로써 프로그램의 구조를 단순하게 유지하는 대신에 입력자료의 편집기능과 염기에 따라 각기 다른 색깔의 문자로 표현하는 기능을 갖도록 작성하였다(Fig. 1). 또한 이미 보고된 유사한 염기서열을 비교용 참고 염기서열(reference sequence)로 이용할 수 있는 기능을 추가하여 실제 염기서열 해독작업에 도움이 되도록 하였으며, 다중 윈도우를 지원하여 작업효율을 높일 수 있도록 하였다.

프로그램의 기본구조는 염기서열을 입력하는 부분, 편집하는 부분, 출력하는 부분, 최대 유사도를 갖도록 정렬하는 부분, 결과를 저장하는 부분, 프로그램 환경을 설정하는 부분으로 구성되어 있으며 각 처리루틴을 세분화하고 확장하였다. 특히 사용자의 입력과 자료의 출력을 편리하게 처리하기 위해서는 효율적인 사용자 인터페이스가 필요하며, 측정결과를 사용자에게 편리한 형식으로 보여주기 위한 여러 화면처리를 위한 루틴을 제작하였고 최근의 운영체제가 제공하는 다양한 그래픽 사용자 환경을 충분히 활용할 수 있도록 하였다.

각 단편의 염기서열과 비교용 참고 염기서열을 입력하는 작업은 동일한 루틴을 사용하며 다양한 형태의 자료를 읽을 수 있도록 하였다. 입력 가능한 자료의 형식(Format)은 현재 폭

넓게 사용되고 있는 FASTA(14), GenBank(5) 형식뿐만이 아니라 ASCII 형식의 파일도 포함되며, 사용자의 확인 없이도 자료의 첫줄을 읽고 해당 형식을 파악할 수 있도록 작성되었다. 현재까지 개발된 대부분의 프로그램이 입력자료에 사용할 수 있는 염기의 종류는 A, G, C, T이며 불확실한 정보는 N으로 처리하고 있다. 그러나 실제로 염기서열 해독실험의 결과를 판독할 때에 정확히 판별할 수 없는 염기도 실제로는 어느 정도 정보를 포함하게 된다. 즉 R(Purine), Y(Pyrimidine), B(C 또는 G 또는 T)처럼 확실한 판정을 내릴 수 없는 염기라도 최소한의 정보는 포함하고 있어 B인 경우 최소한 A가 아님을 알 수 있다. 따라서 불완전한 정보를 단순히 N으로만 표시하면 최소한의 정보조차도 사용할 수가 없으며 잘못된 contig를 구성하는 한 요인이 되기도 한다. 이러한 문제점을 줄이기 위해 본 연구에서 개발된 프로그램의 입력자료에 IUB-IUPAC(International Union of Biochemistry-International Union of Pure and Applied Chemistry)의 염기표기법(7)을 사용할 수 있도록 작성하였으며, 국내에서 개발된 FAP(Fragment Assembly Program)과 XFAP(X-window-based Fragment Assembly Program) 또한 IUB-IUPAC 표기법을 사용하는 것으로 보고된 바 있다(2, 3).

최근에는 자동염기서열 분석장치(automatic sequence analyzer)를 이용하여 염기서열 해독작업을 하는 사례가 많으며 이 경우 각 단편의 3'-말단은 부정확한 정보를 포함하는 경우가 자주 발생한다. 또한 반대 방향의 상보적 핵산(complementary strand)을 대상으로 염기서열 해독작업을 하여 이미 획득한 실험결과를 검증하는 경우도 많다. 따라서 실험자가 입력된 자료를 편집할 수 있는 기능을 추가하였으며, 특정 염기의 수정뿐만이 아니라 특정 염기이후 부분의 일괄 삭제기능과 상보적 염기서열로 변환하는 기능도 추가하였다 또한 편집기능은 단편을 정렬한 후에 얻어진 결과의 수정에도 유용하며, 이미 보고된 염기서열과 한 화면에서 비교가 가능하기 때문에 작업효율을 높일 수 있으며 rRNA 같이 보존적인 유전자(conserved gene)는 해독작업의 오류도 발견할 수 있다. Fig. 1에서는 *Penicillium purprogenum*의 18S rRNA 유전자를 NS3와 NS4 primer(22)로 단편 염기서열을 해독하고 contig를 구성한 후에 GenBank의 *P. marneffei*(Acces-sion No. : AF034197)의 염기서열과 비교하는 사례를 나타내고 있다. NS3 primer로 염기서열을 해독한 단편의 339번째 염기가 C인데 NS4 primer로 해독한 단편과 참고용 *P. marneffei*의 염기는 G이므로 NS3 단편의 3'-말단에서 오류가 발생하였음을 쉽게 알 수 있다.

프로그램 기본환경의 설정 메뉴에서는 화면배색 등 다양한 항목들을 설정하고 저장할 수 있다. 화면바탕색과 IUB-IUPAC의 염기표기법에 해당되는 각 염기의 표현색을 자유롭게 선택할 수 있으며, 견본 염기서열과 조립용 염기서열에 각기 다른 색조합을 사용할 수 있으므로 단편정렬 작업을 할 때에 효율적이고 사용자의 기호에 맞는 그래픽 사용자 환경을 유지할 수 있다. 또한 비교 염기서열로 사용할 파일을 선택할 수 있으며, 단편을 정렬하기 위하여 행렬을 계산할 때에 사용되는 계산식(수식 1, 2)의 염기유사도$S(a_i, b_j)$와 결실벌점 q의 값을 지정할 수 있다. 이들 유사도 및 벌점 항목은 자유롭게 수정이 가능하지만 단편정렬 과정에 큰 영향을 끼치므로 주의하여야 한다. 프로그램에서 제공되는 기본값은 두 종류가 있으며, 두 염기가 서로 모순되는 경우의 벌점과 결실로 인한 공백이 형성될 경우의 벌점에 차이가 있다. 따라서 각 단편의 염기서열 정보의 신뢰도에 따라 염기불일치에 해당되는 벌점과 결실벌점을 조정하면 단편정렬 과정에서 공백이 추가되는 경향을 제어할 수 있다.

단편정렬 방법의 검토

두 단편(fragment)의 염기서열로부터 국부적 상동성(local homology)을 확인하여 최대 유사도(maximum similarity value)를 나타내도록 정렬하는 방법으로 Smith-Waterman의 알고리즘(algorithm)이 잘 알려져 있으며, 컴퓨터 프로그램으로 구현이 가능하다(17). 두 단편의 염기서열이 $A=a_1a_2a_3\cdots\cdots a_N$와 $B=b_1b_2b_3\cdots\cdots b_M$일 경우, $S(a_i, b_j)$는 두 단편의 i번째와 j번째 염기사이의 유사도이며, W_k는 k 길이만큼의 염기가 결실 또는 삽입되어 공백을 형성하였을 경우의 벌점이라고 할 때, 유사도 값이 최대가 되는 부분의 정렬을 찾을 수 있는 Smith-Waterman의 수식은 아래와 같은 행렬로 나타낼 수 있다.

$$H_{k,0}=H_{0,l}=0,\ 0\le k\le n \text{ and } 0\le l\le m \quad (5)$$

$$H_{i,j}=\mathrm{Max}\{H_{i-1,j-1}+s(a_i, b_j),\ P_{i,j},\ Q_{i,j},\ 0\},\ 1\le i\le n \text{ and } 1\le j\le m$$

$$P_{i,j}=\mathrm{Max}_{1\le k\le i}\{H_{i-k,j}-W_k\},\ Q_{i,j}=\mathrm{Max}_{1\le k\le j}\{H_{i,j-k}-W_k\} \quad (6)$$

그러나 Smith-Waterman의 알고리즘이 두 염기서열 사이에서 국부적 상동성를 찾는데 매우 효과적이지만 행렬을 완성하는데 상당히 많은 계산이 필요하며, 특히 긴 염기서열을 비교할 때 컴퓨터 수행시간이 길어지게 된다. 따라서 Smith-Waterman의 방법을 개선할 필요가 있으며 여러 가지 새로운 방법이 제시되었다. 먼저 결실벌점 W_K가 k의 일차함수일 때, 즉 $W_k=ku+v$의 관계를 갖고 있을 때, 수식 (6)의 $P_{i,j}$와 $Q_{i,j}$는 각각 아래와 같이 간단히 표현될 수 있다(8).

$$P_{i,j}=\max_{1\le k\le i}\{H_{i-k,j}-W_k\}=\mathrm{Max}\{H_{i-1,j}-W_1,\ P_{i-1,j}+u\} \quad (7)$$

$$Q_{i,j}=\max_{1\le k\le j}\{H_{i,j-k}-W_k\}=\mathrm{Max}\{H_{i,j-1}-W_1,\ Q_{i,j-1}+u\} \quad (8)$$

새로운 $P_{i,j}$와 $Q_{i,j}$의 계산방법은 비교횟수를 줄임으로써 컴퓨터 작업시간을 상당히 단축 시킬 수 있다. 또한 행렬 H를 계산한 후 최대 값을 갖는 원소를 확인하고 주변 원소를 비교하면서 역추적을 하는 방법 대신에 dynamic programing 방법(19)을 적용하면 행렬 H를 계산함과 동시에 각 원소의 추적 방향을 갖는 행렬 D를 만들어 프로그램의 작업효율을 높일 수 있다.

한편 CAP(Contig Assembly Program)을 개발한 Huang (10)은 두 단편의 정렬시 단일 염기의 삽입이나 결실에 의한 공백의 벌점에 q라는 상수를 지정하고, Max 함수에서 0을 제거함으로써 아래와 같이 더욱 간단하게 변형된 수식을 사용하였다(수식 1, 2). 또한 두 단편이 정렬되는 형태를 단순화하여

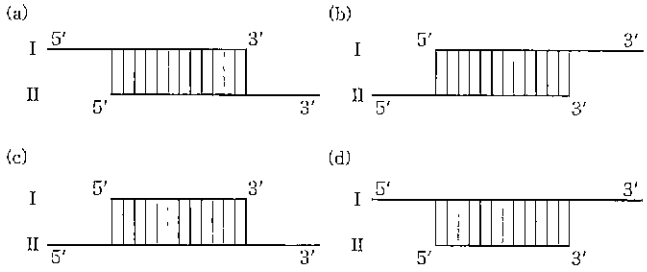

Fig. 2. Four types of local alignments between fragments I and II. (a) A 3' segment of I overlaps a 5' segment of II. (b) A 3' segment of II overlaps 5' segment of I. (c) Fragment I is contained in fragment II. (d) Fragment II is contained in fragment I.

Fig. 2에 제시된 4가지 경우로 해석하였다. 즉 중첩부분은 항상 두 단편중 한 단편의 5' 말단에서 시작하여 한 단편의 3'-말단에서 끝나게 된다. 따라서 Smith-Waterman의 알고리즘처럼 행렬 H의 전체 원소 중에서 최대값을 갖는 원소를 찾을 필요 없이 마지막 행과 열의 원소들만 비교하면 된다(수식 3).

$$H_{i,j}=0,\ \text{if } i=0 \text{ or } j=0 \quad (1)$$

$$H_{i,j}=\text{Max}\{H_{i-1,j-1}+s(a_i, b_j),\ H_{i-1,j}-q,\ H_{i,j-1}-q\},\ \text{if } i>0 \text{ and } i>0 \quad (2)$$

$$\text{Maximal value}=\text{Max}\{H_{i,M},\ H_{N,j}: 1\le i\le N \text{ and } 1\le j\le M\} \quad (3)$$

그러나 수식 (3)과 같이 행렬의 끝 부분에서 선택된 원소로부터 역추적을 하여 두 단편을 정렬하게 되면 단편 말단의 염기서열이 부정확한 경우 올바른 정렬을 기대할 수 없으며, 실제로 실험결과로부터 얻어지는 단편의 염기서열은 각 말단부위에서 부정확한 정보를 포함하는 경우가 많다. 이러한 문제점을 최소화하기 위하여 Huang은 행렬 계산에 사용되는 염기 유사도 $S(a_i, b_j)$와 결실벌점 q의 값을 단편의 중간부위와 말단에 따라 각기 다른 값을 지정하였다. 그러나 실제로 실험오차가 증가하는 말단의 범위는 실험자의 경험 및 실험조건에 따라 다르므로 정보가 부정확한 말단의 범위를 일관되게 적용할 수 없으며, 단편의 중간과 말단 부위에 각각 적용할 염기 유사도와 결실벌점의 값을 적절하게 결정하기 어렵다.

따라서 본 연구에서는 수식 (1)과 (2)를 이용하고 염기유사도 $S(a_i, b_j)$와 결실벌점 q의 값을 말단 부위에서도 동일하게 적용하여 효율적인 계산방법을 구축하였다. 또한 3'-말단의 정보가 부정확한 경우에 최대값을 갖는 원소가 행렬의 마지막 행 또는 열에 위치하지 않는 문제는 최대값을 갖는 원소를 기준으로 하위행렬(submatrix)을 구분하고 하위행렬에서도 독립적으로 단편을 정렬하는 방법을 이용하였다. 즉 행렬 값의 계산과정 중에 최대값을 갖는 원소의 위치를 기억시키며, 최대값을 갖는 원소가 행렬의 마지막 행 또는 열에 위치하지 않을 경우, 수식 (4)에 의하여 하위행렬에서 역추적에 필요한 원소의 위치를 확인하였다.

$$\text{New value}=\text{Max}\{H_{i,M},\ H_{N,j}: k\le i\le N \text{ and } 1\le j\le M\},\ \text{if } H_{k,l}=\text{Maximal value} \quad (4)$$

전체 염기서열이 아래와 같은 유전자로부터 단편 (1)과 단편 (2)의 염기서열 정보를 얻었으나 단편 (1)의 3'-말단 부위에 오류가 발생한 경우에, 본 연구에서 사용된 방법으로 두 단편을 정렬하는 예가 Fig. 3에 제시되어 있다.

…N·N·A·C·T·G·C·C·T·A·C·G·A·A·G·G·C·A·N·N (전체염기서열)
…N·N·A·C·T·G·C·C·T·A·C·G·T·C·C·G—·A (단편 1)
G·C·C·T—·C·G·A·A·G·G·C·A·N·N… (단편 2)

먼저 행렬 H의 첫 번째 행과 첫 번째 열의 모든 원소들을 식 (1)과 같이 0으로 초기화하면, 두 단편에서 각각 처음부터 i번째와 j번째까지의 염기서열을 갖는 부분($A_i=a_1a_2a_3\cdots\cdots a_i$, $B_j=b_1b_2b_3\cdots\cdots b_j$)을 정렬하였을 경우의 최대 유사도 $H_{i,j}$는 수식 (2)에서 계산될 수 있다. 이때 염기의 유사도$S(a_i, b_j)$ 값으로 두 염기가 일치할 경우($a_i=b_j$)에는 2, 틀린 염기일 경우($a_i\neq b_j$)에는 -3을 지정하며, 염기의 결실벌점(deletion penalty) q에는 -2의 값을 지정하였다. 따라서 두 단편의 염기서열이 최대 유사도를 나타내도록 정렬시키려면 먼저 행렬 H에서 최대값을 갖는 원소를 찾아야 한다. 그후 이 원소가 가장 큰 값을 갖도록 기여한 원소들을 순서대로 역추적하면 최대값 정렬을 갖는 부분(segment)의 정보를 알 수 있다. Fig. 3의 전체 행렬 중에서 최대 값을 갖는 원소는 $H_{6,10}=10$이며, 다음 원소를 역추적하기 위해 좌측 열과 위 행에 존재하는 원소들을 살펴보면 $H_{5,9}$가 8이고 두 염기가 일치할 경우의 유사도 2가 더하여 $H_{6,10}$가 10이 되도록 꽌여했음을 쉽게 알 수 있다. 따라서 $H_{6,10}$의 형성에 기여한 원소의 역추적 결과는 $H_{5,9}$이다. 행렬의 원소들을 역추적할 때에 대각선 방향의 이동은 염기의 결실 또는 삽입이 없는 경우이고, 좌측 또는 위쪽으로 이동하는 경우 해당되는 단편의 염기서열에 공백을 추가하게 된다. 또한 최대값을 갖는 원소의 위치가 행렬의 마지막 부분이 아니므로 $H_{7,11}$부터 $H_{12,15}$까지 하위행렬로 구분하고 수식 (4)에 의하여 $H_{7,15}=5$부터 역추적을 시작한다. 따라서 $H_{7,15}$부터 $H_{1,4}$까지 $H_{6,10}$을 경유하도록 역추적을 계속한 후에, 두 단편에서 최대 유사도를 나타내는 부분을 정렬하면 아래와 같다.

…N·N·A·C·T·G·C·C·T·A·C·G·T·C·C·G·A (단편 1)

Sequence (Matrix No.)	(0)	G (1)	C (2)	C (3)	T (4)	C (5)	G (6)	A (7)	A (8)	G (9)	G (10)	C (11)	A (12)
(0)	0	0	0	0	0	0	0	0	0	0	0	0	0
A (1)	0	-2	-2	-2	-2	-2	-2	2	2	0	-2	-2	2
C (2)	0	-2	0	0	-2	0	-2	0	0	-1	-3	0	0
T (3)	0	-2	-2	-2	2	0	-2	-2	-2	-3	-1	-2	-2
G (4)	0	*2*	0	-2	0	-1	2	0	-2	0	-1	-3	-4
C (5)	0	0	*4*	2	0	2	0	-1	-3	-2	-3	1	-1
C (6)	0	-2	2	*6*	4	2	0	-2	-4	-4	-5	-1	-2
T (7)	0	-2	0	4	*8*	6	4	2	0	-2	-4	-3	-4
A (8)	0	-2	-2	2	*6*	*5*	3	6	4	2	0	-2	-1
C (9)	0	-2	0	0	4	*8*	6	4	3	1	-1	2	0
G (10)	0	2	0	-2	2	6	*10*	8	6	5	3	1	-1
T (11)	0	0	1	-3	0	4	8	7	5	3	2	0	-2
C (12)	0	-2	2	1	-1	2	6	5	4	2	0	4	2
C (13)	0	-2	0	4	2	1	4	3	2	1	-1	2	1
G (14)	0	2	0	2	1	-1	3	1	0	4	3	1	-1
A (15)	0	0	-1	0	-1	-2	1	*5*	3	2	1	0	3

Fig. 3. The local homology matrix ($H_{i,j}$) generated from the sequences A–C–T–G–C–C–T–A–C–G–T–C–C–G–A and G–C–C–T–C–G–A–A–G–G–C–A. The segment of maximal similarity is represented by bolded sequence characters. The gray-colored elements indicate the traceback paths from the maximal element ($H_{7,15}=5$) in the last row. The elements surrounded with dot and bold lines are selected by Huang's and our algorithms respectively. The submatrix separated by maximal element ($H^{6,10}=10$) is represented by bold lines.

G·C·C·T·–·C·G·–·–·–·–·A·A·G·G·C·A·N·N
(단편 2)

그러나 동일한 행렬에 단순한 역추적 방법을 적용하면 행렬 중에서 최대 값을 갖는 원소는 $H_{6,10}$=10이지만 수식 (3)에 의해 $H_{7,15}$=5가 출발점이 되며 하위행렬이 구분되지 않았으므로 Fig. 3에서 점선으로 표시된 경로를 따라 $H_{1,10}$=2까지 역추적이 진행되고 정렬결과는 아래와 같다.

···N·N·A·C·T·G·C·C·T·A·C·G·T·C·–·C·G·A (단편 1)
G·C·C·T·C·G·A·A·G·G·C·A·N·N
(단편 2)

따라서 두 단편에서 최대 유사도를 나타내는 부분을 정렬하는데 있어서 본 연구에서 개발된 프로그램이 더욱 효과적이며, 실험과정의 오류로 인하여 말단 부위에서 왜곡된 정보가 증가하는 경우에도 적용이 가능함을 알 수 있다. 실제로 세균 및 *Penicillium* 속의 곰팡이로부터 해독된 16S rRNA와 18S rRNA 유전자의 단편 염기서열을 재구성하는 작업에 프로그램을 사용하였을 때에 효율적인 작업이 가능하였다. 본 프로그램의 실행화일은 인터넷(http://FunFact.kribb.re.kr, http://210.219.43.25/kordic.htm)으로 제공되며 다양한 분자생물학적 연구에서 비전문가도 쉽게 활용할 수 있다.

감사의 말

이 논문은 1998년 충북대학교 학술연구재단 연구비에 의하여 연구되었습니다.

참고문헌

1 **박기정, 김승목, 박찬규, 박용하.** 1995. RNA 분석 프로그램의 개발. *The Microorganisms & Industry.* **21**, 375-382.
2. **이병욱, 박기정, 김승목.** 1998 DNA 염기 서열로부터 contig 구성을 위한 프로그램 XFAP의 개발 *Kor. J. Microbiol.* **34**, 58-63.
3. **이병욱, 박기정, 박완, 박용하.** 1997. DNA 염기서열의 단편 조립 프로그램의 개발. *Kor. J. Appl Microbiol. Biotechnol.* **25**, 560-565.
4. **한경숙, 김홍진.** 1996. 컴퓨터 모델링을 이용한 RNA 분자의 이차구조 예측. 정보과학회논문지 (B). **23**, 75-84.
5. **Benson, D.A., M.S. Boguski, D.J. Lipman, J. Ostell, and B.F. Ouellette.** 1998. GenBank. *Nucl. Acids Res.* **26**, 1-7.
6. **Bonfield, J.K., K.F. Smith, and R. Staden.** 1995. A new DNA sequence assembly program. *Nucl. Acids Res.* **23**, 4992-4999.
7. **Cornish-Bowden, A.** 1985. A nomenclature for incompletely specified bases in nucleic acid sequences: recommendation. *Nucl. Acids Res.* **16**, 3021-3030.
8. **Gotoh, O.** 1982. An improved algorithm for matching biological sequences. *J Mol. Biol.* **162**, 705-708.
9 **Huang, X.** 1996 An improved sequence assembly program. *Genomics* **33**, 21-31.
10 **Huang, X.** 1992 A contig assembly program based on sensitive detection of fragment overlaps. *Genomics* **14**, 18-25.
11. **Lane, D.J., B. Pace, G.J. Olsen, D.A. Stahl, M.L. Sogin, and N.R. Pace.** 1985. Rapid determination of 16S rRNA sequences for phylogenetic analysis. *Proc Natl. Acad. Sci. USA.* **82**. 6955-6959.
12. **Lee, D.H., Y.G. Jo, and S.J. Kim.** 1996. Nonradioactive method to study genetic profiles of natural bacterial communities by PCR-Single-Strand-Conformation Polymorphism. *Appl. Environ. Microbiol.* **62**, 3112-3120.
13. **Lee, D.H., Y.J. Kim, S.T. Lee, and H.S. Kang.** 1986. Rapid plasmid mapping computer program *Kor. J. Microbiol.* **24**, 12-17.
14. **Maniatis, T., A. Jeffrey, and D.G. Kleid.** 1975. Nucleotide sequence of the rightward operator of phage lambda. *Proc. Natl. Acad. Sci. USA* **72**, 1184-1188
15 **Marmur, J. and P. Doty.** 1962 Determination of base composition of deoxyribonucleic acid from its thermal denaturation temperature. *J. Mol Biol.* **5**, 109-118.
16. **Pearson, W.R. and D.J. Lipman.** 1988. Improved tools for biological sequence comparison. *Proc. Natl. Acad Sci. USA.* **85**, 2444-2449.
17. **Peltola, H., H. Soderlund, and E. Ukkonen.** 1984. SEQAID: a DNA sequence assembling program based on a mathematical model. *Nucl. Acids Res.* **12**, 307-321.
18. **Repetto, M., A. Ballabio, and M. Zollo.** 1997. A method to direct sequence cosmid LAWRIST16 clones. *DNA Seq* **7**, 229-233.
19. **Smith, T.F. and M.S. Waterman.** 1981. Identification of common molecular subsequences. *J. Mol. Biol.* **147**, 195-197.
20 **Staden, R.** 1980. A new computer method for the storage and manipulation of DNA gel reading data. *Nucl. Acids Res.* **8**, 3673-3694.
21. **Waterman, M.S. and T.H. Byers.** 1985. A dynamic programming algorithm to find all solutions in a neighborhood of the optimum. *Mathematical Biosciences* **77**, 179-188.
22. **White, T.J., T. Bruns, S. Lee, and J. Taylor.** 1990. Amplification and direct sequencing of fungal ribosomal RNA genes for phylogenetics, p 315-322. *In* M.A. Innis, D.H. Gelfand, J.J. Sninsky, and T.J White (ed.), PCR Protocols: a guide to methods and applications Academic Press, Inc., Los Angeles, California.

(Received March 22, 1999/Accepted May 8, 1999)

ABSTRACT : Development of Contig Assembly Program for Nucleotide Sequencing

Dong-Hun Lee* (Division of Life Sciences, Chungbuk National University, Cheongju 361-763, Korea)

An effective computer program for assembling fragments in DNA sequencing has been developed. The program, called SeqEditor (Sequence Editor), is usable on the personal computer systems of MS-Windows which is the most popular operating system in Korea. It can read several sequence file formats such as GenBank, FASTA, and ASCII. In the SeqEditor program, a dynamic programming algorithm is applied to compute the maximal-scoring overlapping alignment between each pair of fragments. A novel feature of the program is that SeqEditor implements interactive operation with a graphical user interface. The performance tests of the program on fragment data from 16S and 18S rDNA sequencing projects produced satisfactory results. This program may be useful to a person who has work of time with large-scale DNA sequencing projects.