서열의 효율적인 유사성 분석 방법

김양석

포항공과대학교 생물학 정보 센터

1. 들어가는 말

급속한 분자 생물학과 생물정보학(Bioinformatics)의 발달로 인해 많은 유전자 및 단백질 데이터베이스가 구축되었고, 포함된 정보의 양은 기하 급수적으로 증가하는 추세에 있다. 생물학자들은 구축된 데이터베이스의 검색을 통해 많은 정보를 얻을 수 있으며 특히 유사성 검색을 통한 서열의 특성 분석 및 기능 예측은 분자 생물학자들의 가장 중요한 연구 수단중의 하나로 인식되고 있다. 하지만 데이터베이스와 검색 프로그램들은 대부분 수학자나 전산학자들에 의해 개발되어 검색 프로그램들의 특징이나 검색 조건들이 생물학자들이 이해하기는 쉽지 않다. 본 review 에서는 생물학자들이 가장 많이 사용하는 일반 유사성 검색 프로그램들에 대한 소개와 효율적인 서열 분석을 위한 검색 프로그램들의 사용 방법에 대해 소개한다.

2. 효율적인 유사성 검색을 위해 고려할 점

2.1 단백질과 유전자 서열의 비교

일반적으로 실험실에서 새로이 밝히는 서열은 DNA sequencing 을 통해 밝혀낸 염기 서열이지만 유사성 분석을 위해서는 밝혀진 유전자 서열을 translation 시켜 단백질 서열로 비교하는 것이 훨씬 더 정확하다. 그 이유는 다음과 같다.

(1) 유전자 서열은 4 종류(A,C,G,T)로 구성되어 있지만 단백질 서열은 20 개의 아미노산 서열로 구성되어 있다. 따라서 우연히 두 residue 가 일치할 확률은 단백질은 1/20 이지만 유전자의 경우는 1/4 이다. 따라서 두 서열을 비교할 때 단백질 서열이 일치하는 것이 확률적으로 더 큰 의미를 가지게 된다.

(2) 유전자는 codon degeneracy 와 각 개체별로 codon preference 를 가지고 있다. 이러한 유전자의 coding 특성에 의해 다른 유전자가 같은 아미노산을 coding 할 수 있고, 이것은 단백질 level 에서 같은 서열을 가짐에도 불구하고 유전자 서열에서는 많은 차이가 날 수 있음을 의미하고 진화적으로 멀리 떨어진 개체의 유전자 서열 비교는 큰 의미가 없음을 알 수 있다.

(3) 현재 개발된 대부분의 검색 프로그램들은 단백질 검색을 목적으로 개발되었다. 뒷장에서 설명할 scoring matrix 등 검색에 중요한 영향을 미치는 factor 들은 단백질 검색에 적절하게 구성되어 있다.

(4) 단백질 서열 데이터의 크기가 유전자에 비해 훨씬 작다. 현재 GenBank 등 3 대 유전자 데이터베이스에 등록된 유전자의 종류는 200 만 건에 육박하고 있지만 Swiss-Prot 에 등록된 단백질의 개수는 10 만 건에 미치지 못하고 있다. 따라서 단백질 검색이 유전자 검색보다 빨리 수행된다.

2.2 공통된 검색 옵션

Scoring matirx 와 gap penalty 는 개발된 유사성 분석 프로그램들에 공통적으로 사용되고 검색 결과에 많은 영향을 미친다.

2.2.1. Scoring matrix

아래의 두개의 아미노산 배열을 고려해 보자. 두개의 배열에서 공통되는 residue 의 수로 점수를 배긴다고 한다면 두 배열은 9 개 중에 5 개가 일치하므로 같은 점수일 것이다.

```
a) TTYGAPPWCS          b) TTYGAPPWCS
   TGYAPPPWS              TGYAPPPWS
   *  ***  *             * *  ***
```

그러나 배열 a)는 상대적으로 보편적인 residues (A, P, S, T) 만을 보존하고 있지만 배열 b)에는 W 와 T 같은 덜 보편적인 residue 들이 보존되어 있다. 즉 적절한 배열을 위해서는 각 아미노산들의 진화적, 화학적, 물리적 성질들을 고려하여야 함을 알 수 있다.

Scoring matrix 는 두 서열을 비교할 때 각각의 아미노산이나 염기들이 일치 혹은 치환될 확률을 미리 계산하여 만든 표이다. 현재 사용되는 scoring matrix 의 종류는 다음과 같다.

가. 염기 측정 (nucleotide scoring)에 쓰이는 scoring matrix 들 (scoring matrix)

DNA 배열에 대한 scoring matrix 은 상대적으로 간단하다. 염기의 경우 세가지 정도의 matrix 가 주로 사용되고 있다.

(1) Identity matrix (similarity)

	A	T	C	G
A	1	0	0	0
T	0	1	0	0
C	0	0	1	0
G	0	0	0	1

(2) BLAST matrix (similarity)

	A	T	C	G
A	-5	-4	-4	-4
T	-4	5	-4	-4
C	-4	-4	5	-4
G	-4	-4	-4	5

(3) Transition/Transversion Matrix

	A	T	C	G
A	0	5	5	1

```
T 5 0 1 5
C 5 1 0 5
G 1 5 5 0
```

나. 단백질 측정 (protein scoring)에 쓰이는 scoring matrix 들

단백질은 20 종류의 아미노산으로 구성되어 있으므로 아미노산의 scoring matrix 는 20×20 행렬로 표현할 수 있다. 아미노산의 scoring matrix 는 단백질 비교의 직접적인 기준이 되므로, 현재에도 아미노산의 진화적, 물리적, 화학적 성질을 고려한 여러 종류의 행렬들이 개발되고 있다. 그 중 genetic code matrix, physical /chemical characteristics 를 이용한 matrix 와 현재 일반적으로 많이 이용되는 PAM 과 BLUSUM 의 특징은 다음과 같다.

- Genetic code Matrix: 한 개의 아미노산이 다른 아미노산으로 바뀌는데 필요한 최소한의 염기 서열의 개수 계산

- Physical/chemical characteristics: 서로 다른 두 아미노산의 물리적, 화학적 성질의 유사성을 이용하여 점수를 부여한 방법 예) hydrophobicity matrix

-Dayhoff Mutation Data Matrix:

Dayhoff 등에 의해 개발된 mutation data matrix (이하 MDM)은 현재 일반적으로 가장 많이 쓰이고 있는 scoring matrix 중의 하나이다 (Dayhoff et.al., 1978). 1978 년에 처음 발표되었을 때는 당시에 알려진 단백질 서열들과 그 서열들에서 유추된 ancestral 서열들로부터 얻은 400 개의 accepted point mutation 을 이용하여 MDM 이 제작되었다. 이후 여러 개체들의 서열들이 밝혀짐에 따라 MDM 은 계속 확장 되어 1980 년에는 71 개의 연관된 group (서열이 85% 이상 동일한 group)들로부터 얻은 1600 개의 accepted point mutation 들을 근거로 MDM 이 제작되었다.

단백질 서열의 mutation 에 관한 Dayhoff model 은 단백질 치환에 관한 Markovian 모델 (model)을 근거로 한다. Markovian 모델은 한 단백질 내에서의 어떤 특정 위치의 mutation 은 다른 위치의 mutation 과 무관하다는 것을 전제로 한다.

Markovian 모델내에서 MDM 은 한 단위의 진화적 변화(one unit of evolutionary change) 동안 아미노산 A 가 아미노산 B 로 치환될 확률을 계산한 transition probability matrix 로부터 유도 된다. 행렬의 대각선의 값들은 각각의 아미노산이 변하지 않을 확률을 나타낸다. 즉 대각선에 위치한 값들의 합은 주어진 진화 기간(represented evolutionary interval)동안 amino acid 가 변하지 않을 확률을 나타내게 된다. Dayhoff derivation 에서는 대각선의 값들의 합이 99%가 되게 probability matrix 를 조정하였다. 그러므로 probability matrix 에서의 진화 단위(unit of evolution)는 100 개의 site 중 1 개의 site 에서 accepted amino acid substitution 이 일어날 확률에 해당한다. (1 PAM unit). 진화 단위에는 시간적 개념이 전혀 고려되고 있지 않음을 유의해야 한다.

MDM 의 중요한 단점은 단백질의 각 site 에서 mutation 이 일어날 확률은 일정하지 않다는 것이다. 현대 분자 생물학에 있어서 단백질 내에서 각 site 에 따라 mutation 이 일어날 확률은 다르다는 사실은 잘 알려져 있다. 그러므로 각 site 에서의 mutation 의 확률을 동일하게 고려한 Dayhoff 의 모델은 한계를 가지고 있다.

- BLOSUM (BLOks SUbstitution Matrix):

1991 년에 Altschul 등에 의해 발표된 BLOSUM 은 현재 BLAST 등의 검색에 제공되며 PAM 과 함께 가장 많이 쓰이는 scoring matrix 의 한 종류이다. BLOSUM 은 Block database 로부터 개발된 것으로, Block 데이터베이스는 아미

노산 서열 중 다른 부분에 비해 굉장히 보존된 (conserved) 부분만을 모아 만든 데이터베이스이다. 이중 일부는 어떤 기능을 가진 motif 로 알려져 있다.

PAM 이 연관된 서열들과 유추된 서열을 적절히 배열한 후 치환 확률을 구하는 반면 BLOSUM 은 block 내에서 아미노산들을 배열한 후 각각의 아미노산들이 짝(pair)을 이루는 확률을 관찰해서 치환 확률을 구한 것이다.

연속적인 치환 행렬을 만들기 위해 서열들을 각각의 block 에 clustering 을 시키고 clustering percentage 는 각각의 group 들에 포함시키기 위한 서열들의 최소한의 일치성 (identity)으로 정의한다. 예를 들면 clustering percentage 가 35%라면 임의의 서열 A 와 B 를 배열시켰을 때 적어도 35% 이상의 identity 를 가지고 있을 때 같은 group 에 포함시키고 BLOSUM35 로 정의한다. 또한 임의의 서열 C 가 A 와 B 둘 중 하나와 35 % 이상의 identity 를 가질 경우에 또한 같은 group 에 포함 시킨다. 각각의 배열된 아미노산 서열들의 pair 들의 개수를 센 후 서열 A,B,C 가 각각 차지하는 비중의 평균을 계산하여 scoring matrix 값들을 구한다.

2.2.2. Gap penalties

Gap penalty 는 삽입 혹은 삭제에 의해 생기는 gap 에 얼마의 감점 (penalty)를 줄 것인가를 정하는 것이다. Gap penalty 를 정확하게 계산 할 수는 없지만 여러 가지 경험적 사실을 통해 −10, −2 에서 −14, −4 정도가 적당하다고 한다. 첫 번째 값은 gap 이 처음 생길 때 주는 감점이고, 두 번째 값은 그 다음에 생기는 연속적인 gap 에 대한 감점이다. 예를 들면 두 개의 서열 사이에 4 개의 gap 이 있고, −10, −2 의 값을 적용하면 전체 gap penalty 는 $-10+3\times(-2)=-16$ 이 된다. 이렇게 다른 값을 적용하는 이유는 진화상에서 처음 gap 이 생기기는 힘들지만 그 이후 연속적으로 생기는 gap 은 처음에 비해 쉽게 생길 수 있기 때문이다. 큰 gap penalty (예를 들면 −14, −4)는 partial sequence (EST 같은)의 비교에 적당하다. 사용자는 gap penalty 를 조정함으로써 sensitivity 를 조절 할 수 있다. 예를 들면 FASTA 검색에서 expectation value 가 0.2 보다 큰 연관성이 거의 없는 서열들이 결과로 출력되었을 때 gap penalty 의 값을 올림으로서 이런 서열들을 제거 해 나갈 수 있다.

2.3. 검색 프로그램

초기의 유사성 검색 도구들은 비교할 서열들의 전체 길이에 대한 "포괄적인 (global)" 유사성 점수를 계산하였다. 하지만 초기에 개발된 이러한 방법들은 진화적으로 거리가 먼 서열들의 비교에는 적당하지 않으므로 현재는 유사성 검색보다는 계통도 작성에 사용된다. 최근 개발된 검색 프로그램들은 이러한 결점을 보안하여 "지역적인 (local)" 유사성 검색 알고리듬을 구현한다. 가장 광범위하게 사용되는 검색 프로그램들로는 Smith−Waterman 방법을 이용한 검색 (Smith and Waterman, 1981), FASTA (Pearson and Lipman, 1988), BLAST (Altschul et. al.,1990) 등이 있다.

2.3.1.Smith−Waterman 방법을 이용한 검색

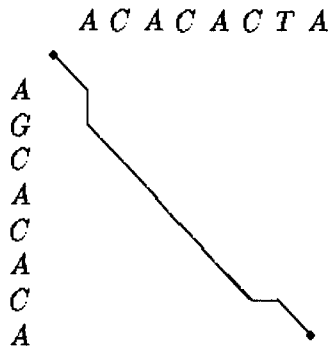
Smith−Waterman 방법은 전산학에서 많이 쓰이는 dynamic programming 기법을 서열 검색에 이용한 방법이다. Dynamic programming 은 전산학의 알고리즘 중의 한 유형으로, 일단 가장 간단한 해답을 알고 있고, 그 해답을 이용해 점점 더 전체로 확장 시켜 나가면서 마지막 대답을 얻는 경우에 이용되는 알고리즘이다. 즉 가장 간단한 s 와 t 의 첫 번째 서열의 적절한 배열에 대한 해답을 쉽게 알 수 있고, 그 값을 근거로 하여 해답을 점점 확장 시켜 전체 서열의 적절한 배열을 구하는 것이다.

다음 두개의 서열로 예를 들어보자.

s= AGCACAGA, t=ACACACTA

여기서 s 와 t 를 각각 행렬의 축으로 하고 unit cost model(일치=0, 치환, 삽입, 삭제=1)을 적용하여 각각의 행렬의 항들을 채워나가면 다음과 같은 행렬을 만들 수 있다.

		t→ A	C	A	C	A	C	T	A
	0	1	2	3	4	5	6	7	8
s↓ A	1	0	1	2	3	4	5	6	7
G	2	1	1	2	3	4	5	6	7
C	3	2	1	2	2	3	4	5	6
A	4	3	2	1	2	2	3	4	5
C	5	4	3	2	1	2	2	3	4
A	6	5	4	3	2	1	2	3	3
C	7	6	5	4	3	2	1	2	3
A	8	7	6	5	4	3	2	2	2



(그림 1) Dynamic programming 을 이용한 서열의 배열

위의 오른쪽 그림은 거리 행렬 (distance matrix)에서 최소값을 나타내는 행로를 표시 한 것이다. 서열 s 에 대하여 대각선은 일치 혹은 치환을, 수평선은 삽입을, 수직선을 삭제를 나타낸다.

대각선에 의해 표시된 행로에 의해 s 와 t 를 배열하면 다음과 같다.

s=AGCACAC-A
t=A-CACACTA

때때로 적절한 배열을 위한 최소한의 행로는 단 한 개가 아니고 여러 개가 될 수 있다.

Dynamic Programming 을 이용하여 임의의 서열과 데이터 베이스에 저장된 서열들을 비교하는 방법을 Smith-Waterman 알고리즘이라 한다. 현재 이 알고리즘을 이용하여 유사성 검색 서비스를 제공하는 곳은 EBI 의 BLITZ 와 Bic-sw Database Searches 가 있고 Weizmann Institute of Science 의 BIOCCELERATOR 는 전자 우편을 통해 결과를 제공하는 서비스를 한다.

Bic-sw 의 서비스 페이지 (http://www2.ebi.ac.uk/Bic-sw/)에 가서 서열을 입력하면 WEB 혹은 e-mail 을 통해 검색 결과를 보내준다. Bic-sw 는 interactive access 를 요구해도 일이 바쁘면 자동적으로 전자 우편으로 보내주기 때문에 access mode 에 관계없이 반드시 전자 우편 주소를 기입해야 한다.

검색 파라미터는 다음과 같다.

- YOUR EMAIL : 검색 결과를 받기 위한 전자 우편 주소를 입력하는 곳
- SEARCH TITLE : 검색 제목을 적는 곳
- RESULT: 결과를 바로 볼 것인가 전자 우편을 통해 받을 것인가를 정하는 곳

- DATABASE : Bic-sw 는 다음의 데이터베이스를 제공한다.

Swall	non-redundant protein database
Swissprot	SWISS-PROT protein database
Swnew	updated to SWISS-PROT
Trembl	TREMBL (Translated EMBL)
Tremblnew	TREMBLNEW (Translated EMBL updates)
Swall	SWISS-PROT + TREMBL + SWISSNEW + TREMBLENEW

(표 1) Bic-sw 에서 제공하는 데이터베이스

- GAPWEIGHT : 배열을 할 때 첫 번째 생기는 gap 에 대한 감점을 정하는 곳으로 기본값은 15 이다.
- LENWEIGHT : 배열을 할 때 연속적으로 생기는 gap 에 대한 감점을 정하는 곳이다. 기본값은 1 이다.
- QUERYGAP : 입력한 서열에서 첫 번째 생기는 gap 에 대한 감점을 정할 수 있다.
- QUERYLEN : 입력한 서열에서 연속적으로 생기는 gap 에 대한 감점을 정할 수 있다.
- MATRIX : scoring matrix 을 정할 수 있다. 기본값은 BLOSUM62 이다.
- SHOW NUMBER OF ALIGNMENTS : 결과 출력파일에서 배열해서 보여주는 서열들의 수를 정할 수 있다.
- SHOW NUMBER OF SCORES : 결과 출력 파일에서 상위 몇 개까지의 score 를 보여주는 가를 정할 수 있다.
- LOWEST Z-SCORE MIN LIST TO REPORT : z-score 가 지정한 값 이상이 되는 list 만 보여주게 정할 수 있다.
- SHOW MINIMUM QUALITY SCORE : 출력 파일에 쓰여지는 결과의 minimum quality 를 정할 수 있다.
- NORMALISATION TYPE : Z score 를 길이에 따라 normalization 하는 방법을 정할 수 있다. LOG 는 logarithmic normalization 을 의미하고 z-score 와 E-value 를 알고 싶으면 stat 를 선택한다. 기본값은 normalization 을 하지 않는 것이다.
- DO AVERAGING OF SCORES : Sequence composition 에 따른 quality score 를 조정할 수 있는 곳이다.
- ORDER OUTPUT BY OVERLAPS QUALITY : overlap option 을 주면 결과에서 alignment 할 때 overlap 이 생기는 부분에 감점을 주게 된다.

2.3.2. FASTA

FASTA 는 임의의 서열과 유사성을 가진 서열을 서열 데이터베이스로부터 찾는 프로그램이다 (http://www2.ebi.ac.uk/fasta3/). FASTA 는 단백질 서열간의 비교를 위해 제작되었지만 염기 서열간의 비교도 가능하다. 특히 TFASTA 의 경우 입력한 단백질 서열과 염기 서열 데이터베이스 간의 비교도 가능하다. 즉 염기 서열 데이터베이스를 6 frame 으로 translation 하여 입력한 단백질 서열과 비교하는데 이 기능은 임의의 단백질 서열과 EST 데이터베이스를 검색하는데 좋은 방법으로 알려져 있다.

가. 알고리즘

FASTA 는 우선 두 서열간의 dot blot 을 그림으로서 비교를 시작한다. Dot blot 에서 일치하는 가진 부분은 대각선으로 표시하고 그려진 대각선들의 합을 계산한다. Smith-Waterman 방법과 FASTA 의 가장 큰 차이점은 FASTA 는 데이터

베이스에 있는 모든 서열들과 dot matrix 를 그리지 않고 대신 FASTA 는 "word"를 기반으로 한 방법을 이용한다는 점이다.

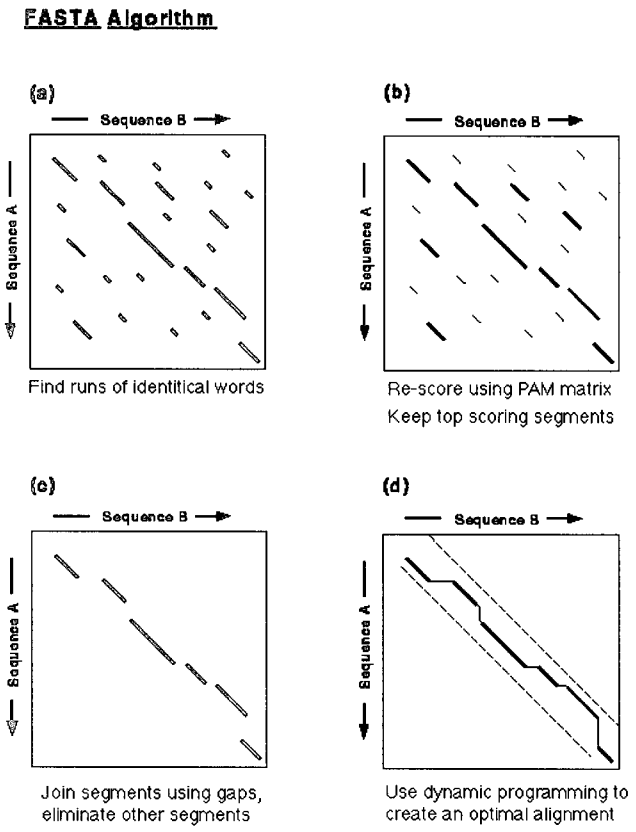
FASTA 가 적절한 서열을 찾아내는 방법은 다음과 같다. (그림 2)

(1) FASTA 는 입력한 서열로부터 한개 (ktup=1) 혹은 두개 (ktup=2)의 단백질 서열 (혹은 3 개 혹은 6 개의 염기서열)로 이루어진 "단어 (ktup)" 들의 조합을 만든다. 그리고 데이터베이스의 임의의 한 서열에서 각 단어들과 일치하는 단어들을 찾아내어 각각의 단어들을 연결하는 대각선을 만든다. 물론 이때 중복된 단어들은 제거한다.

(2) 점수가 높은 대각선 부분 10 개를 선택해 PAM250 과 같은 치환 행렬(replacement matrix)을 이용하여 값(score)들을 다시 계산한다. 이때 가장 큰 값을 가진 부분을 "init1", gap 을 허용하여 몇몇의 high-scoring 대각선 부분들을 합치고 가장 높은 점수를 initn 이라 정의한다.

(3) initn 이 높은 서열들을 선택하고 Smith-Waterman 알고리즘을 이용하여 두 서열을 최적화 배열 (optimal alignment) 하고 값(score)를 계산한다. 이 값을 optimized score (opt.)로 정의한다.

위의 방법에서 단어의 크기(ktup)를 1(한 개의 단백질을 1 개의 단어로 정의)로 하는 것이 비교 속도는 느리지만 훨씬 더 정확하게 비교하는 방법이라는 것을 알 수 있다 (FASTA 에서 기본 단어 값은 2 로 되어 있다).



(그림 2) FASTA 검색 알고리듬

나. FASTA 의 입력 형식

FASTA 는 자체적으로 입력한 서열이 염기 인지 단백질인지를 판단한다. 판단 기준은 입력한 서열 중 A,C,G,T 가 전체 서열의 85% 이상이 되면 염기로 판단하고 그렇지 않으면 단백질로 판단한다. FASTA 는 standard text format sequence file 을 이용한다. 첫 줄은 '>'나 ';'를 쓴 후 comment 를 넣을 수 있다. 그 다음 줄부터 서열을 입력하면 된다. FASTA 는 특수문자나 빈칸, tab 등은 무시하고 single letter amino acid codes 를 인식한다.

예) > Brassica napus BTH1
AREGTKMQSLGGIRSWPATWRTTTASMTTTTTTESVRKVAQVLTVAGSDSGAGAGIQADI
KVCAARGVYCASVKTAVKAKNTRAVQSVHLLPPDSVSEQLKSVLSDFEVDVVKTGMLPS
PEIVEVLLQNLSEYPVRALVVDPVMVSTSGHVLAGSSILSIFRERLLPLADIITPNVKE
ASALLGGVRIQTVAEMRSAAKSLHQMGPRFVLVKGGDLPDSSDSVDVYFDGNEFHELHS
PRIATRNTHGTGCTLASCIAAELAKGSNMLSAVKVAKRFVDSALNYSKDIVIGSGMQGP
FDHFLSLKDPQSYRQSTFKPDDLFLYAVTDSRMNKKWNRSIVDAVKAAIEGGATIIQLR
EKEAETREFLEEAKSCVDICRSNGVCLLINDRFDIAIALDADGVHVGQSDMPVDLVRSL
LGPDKIIGVSCKTQEQAHQAWKDGADYIGSGGVFPTNTKANNRTIGLDGLREVCKASKL
PVVAIGGIGISNAESVMRIGEPNLKGVAVVSALFDQECVLTQAKKLHKTLTESKREH

다. FASTA program 의 종류

FASTA	염기 서열 혹은 단백질 서열간의 유사성 검사
TFASTA	입력한 단백질 서열과 데이터베이스의 염기 서열을 translation 시킨 후 유사성 검사
LFASTA	두 단백질 혹은 염기 서열의 부분 유사성 검색(compare local similarity)을 수행한 후 부분 서열 배열(local sequence alignment)의 결과를 보여줌
PFASTA	두 서열의 부분 유사성 검색 후 부분 서열 배열의 결과를 그림으로 보여줌

(표 2) FASTA program 의 종류

라. FASTA3.0

가장 최근에 나온 FASTA version 으로 서비스 페이지 (http://www2.ebi.ac.uk/ fasta3/)에 가서 서열을 입력하시면 검색을 수행 할 수 있다. FASTA 는 서열을 입력하면 자동으로 염기 서열인지 단백질 서열인지를 판단한다. 즉 전체 서열 중 ACGT 의 서열이 85 % 이상을 차지하면 염기 서열로, 그렇지 않은 경우에는 단백질 서열로 판단한다. 또한 많은 검색 파라미터를 제공하는데 그 중 가장 중요한 값은 ktup 이다. FASTA 에서 염기인 경우 6, 단백질인 경우 2 가 기본값으로 되어 있다. 처음 FASTA 가 개발되었을 경우 PAM 계열의 scoring matrix 밖에 제공되지 않았지만 최근 version 의 경우에는 BLOSUM 계열도 제공하고 있어 BLAST 에 비해 더 좋은 sensitivity 를 가진 것으로 보고되고 있다.

검색 파라미터는 다음과 같다.

YOUR EMAIL, SEARCH TITLE, RESULT, DATABASE 등의 옵션은 Bic-sw 와 동일하다. 단 DATABASE 의 경우 유전자 서열 데이터베이스를 제공하지 않는 Bic-sw 에 비해 더 많은 종류를 제공하며 다음과 같다.

Swall	SWALL Non-Redundant Protein sequence database Swissprot+Trembl+TremblNew	
Swissprot	SWISS-PROT Protein Database	
Swnew	Updates to SWISS-PROT	
Trembl	TREMBL (Translated EMBL)	
Tremblnew	TREMBLNEW	
EMBL	The EMBL Database	Non Interactive
EFUN	EMBL Fungi	
EINV	EMBL Invertebrates	
EHUM	EMBL Human	
EMAM	EMBL Mammalian	
EORG	EMBL Organelles	
EPHG	EMBL Phages	
EPLN	EMBL Plants	
EPRO	EMBL Prokaryote	
EROD	EMBL Rodents	
ESTS	EMBL STSs	
ESYN	EMBL Synthetic	
EUNA	EMBL Unclassified	
EVRL	EMBL Viral	
EVRT	EMBL Vertebrates	
EEST	EMBL ESTs	
EGSS	EMBL Genome Survey Sequences	
EHTG	EMBL High Throughput Genome Sequences	
EMNEW	EMBL New (Updates)	
EMALL	EMBL + EMBL New (Updates)	Non Interactive

(표 3) FASTA 에서 제공하는 데이터베이스

MATRIX: 검색에 사용되는 scoring matrix 을 정할 수 있다. 기본 matrix 는 BLOSUM62 이다.
GAP PENALTIES: Bic-sw 의 경우와 동일하다. GAPOPEN 은 Bic-sw 의 GAPWEIGHT, GAPEXT 는 LINWEIHT 에 해당한다. 주어진 기본값은 다음과 같다.

	DNA	PROTEIN
GAPOPEN	-16	-12
GAPEXT	-4	-2

(표 4) FASTA 의 gap penalty

- SCORES & ALIGNMENTS : Bic-sw 의 "show number of alignment"와 "show number of scores"와 동일한 옵션이다.
- KTUP/WORDSIZE : 위에서 설명한 KTUP 을 결정할 수 있다.
- HISTOGRAM : yes 를 선택하면 결과에 histogram 을 출력한다.
- DNA STRAND : DNA 의 경우 어떤 strand 를 검색할 것인가를 결정 할 수 있다. 기본값은 upper strand 만 검색하게 되어있고 옵션 선택에 따라 bottom 도 할 수 있다.

마. FASTA 의 검색 결과 (output) 분석

FASTA 는 E()-value (expectation of significance)를 계산한다. E()-value 는 결과에 나온 서열이 query 서열과 우연히 배열해 특정 score 이상을 가지는 확률을 뜻한다. 결과에 나온 서열이 생물학적으로 의미가 있다면 E()-value 는 상대적으로 작은 값을 갖는다. 결과에서 보여주는 히스토그램은 데이터베이스의 서열들의 z-score 들의 분포를 보여준다. Z-score 는 optimal score 에 서열의 길이를 계산하여 normalization 한 것으로 서열의 길이에 영향을 받지 않는 값이다. 의미 있는 서열인 경우 Z-score 는 가능한 한 큰 값을 가진다. Z-score 와 expect value 의 그래프는 입력한 서열과 데이터베이스의 서열들이 임의로 유사성을 나타내는 것과 주목할 만한 유사성을 가진 서열들의 구별 기준을 제시해 준다.

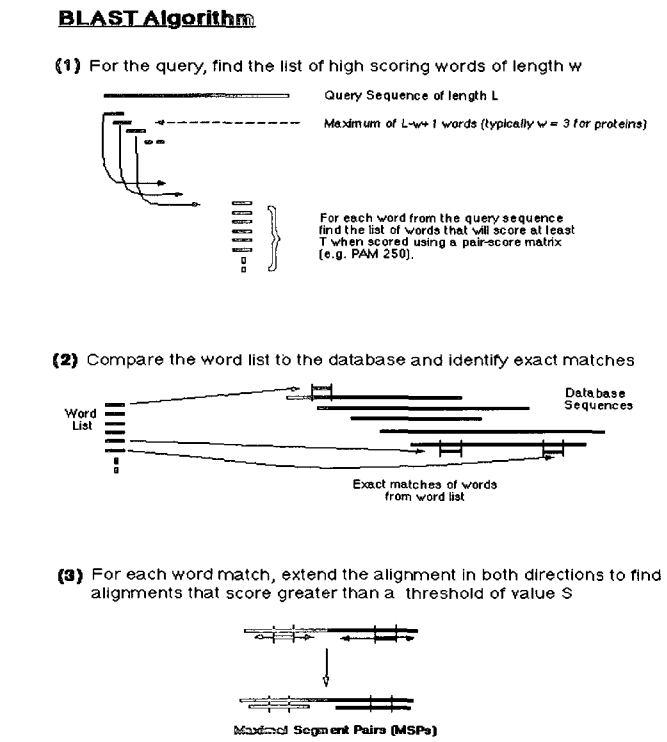
2.3.3. BLAST

BLAST(Basic Local Alignment Search Tool)은 NCBI/GenBank 에서 개발된 유사성 검색 프로그램이다 (http://www.ncbi.nlm.nih.gov/BLAST).

가. 알고리즘

BLAST 는 FASTA 와 마찬가지로 "Word Based Method"를 이용한다. 하지만 FASTA 와는 달리 별도의 pre-formatted 검색 데이터베이스를 필요로 한다. 실제 유사성 검색 과정은 다음과 같다.

(1) 우선 검색을 query 서열로부터 3 개의 단백질 혹은 11 개의 염기로 이루어진 단어들과 T 이상의 점수를 가지는 조합을 만든다. 만들어진 조합들을 각각 서열 데이터베이스의 서열들과 비교한다.
(2) 만약 각각의 단어 조합들과 같은 서열이 서열 database 에서 발견이 되면 BLAST 는 옆 단어들로 유사성 검색을 확장 시켜 나간다. 이때 gap 은 허용하지 않는다.
(3) 확장을 마친 후 데이터베이스 서열 중 일정 값 이상의 HSP(High-scoring Segment Pairs)를 가진 서열들을 추출하고 이때 중복되지 않는 각각의 HSP 들은 통계적인 test 를 거쳐 연결한다.



(그림 3) BLAST algorithm

나. 참고 사항

BLAST 에서 기본적으로 제공하는 서열 데이터베이스(non-redundant, nr)에는 EST 데이터베이스가 포함되어 있지 않다. BLAST 는 query 서열과 gap 없이 일정 값 이상의 HSP 를 기록하지 못하는 서열들을 미리 제거한다. 그래서 FASTA 에 비해 훨씬 비교 속도가 빠르다. 하지만 두 서열이 특정 부분이 높은 일치성을 가지고 있지는 않지만 대부분의 서열에서 유사성을 가지고 있는 경우에는 BLAST 가 검색을 해 낼 수 없다.

또 다른 BLAST 의 단점은 잘 보존되어있으나 큰 의미가 없는 서열들의 부분에 민감하다는 것이다. 즉 short repeat sequence 나 특정한 residue 들이 많이 존재하는 서열 (GC 혹은 AT rich)들이 그 예가 될 수 있는데 이런 서열들을 query 서열로 이용하였을 경우 많은 중요하지 않은 서열들이 결과로 나오게 된다. 이런 결과들을 피하기 위해 BLAST 는 filtering 하는 기능을 기본값으로 가지고 있다. 결국 repeat sequence 같은 것들은 검색하기 이전에 제거된다는 사실을 기억해야 한다.

FASTA 와 마찬가지로 BLAST 도 단백질 서열을 위해 개발된 프로그램이다. 염기 서열의 검색이 가능하지만 sensitivity 가 떨어지므로 염기 서열로 염기 데이터베이스를 검색해 진화적으로 떨어져있는 서열을 찾고자 한다면 FASTA 를 사용하는 게 낫다.

다. BLAST 의 종류

Blastp	단백질 서열간의 비교
Blastn	염기 서열간의 비교
Blastx	입력한 염기 서열을 6 개의 frame 으로 변환 후 단백질 서열 데이터베이스와 비교
Tblastn	염기서열 database 를 6 frame 으로 변환 후 입력한 단백질 서열과 비교
Tblastx	입력한 염기 서열과 염기서열 database 를 모두 6 frame 으로 변환 후 비교

(표 5) BLAST program 의 종류

라. 검색 파라미터들

H, HISTOGRAM: 검색 후 결과에 histogram 을 포함여부를 결정하는 옵션으로 기본값은 포함한다.

V, DESCRIPTIONS: 결과에서 보여주는 유사성을 가진 서열들의 갯수를 정한다. 기본값은 100 개의 유사성을 가진 서열들에 대한 간단한 정보를 결과에서 보여준다.

B, ALIGNMENT: 검색 결과 후 배열을 보여주는 서열들의 갯수로 기본값은 50 개이다.

E, EXPECT: 두 서열의 statistical significance threshold 값이다. 즉 어느 정도 이상의 값을 가져야 두 서열이 유사하다고 정의할 수 있는가에 대한 값을 정하는 옵션으로 기본값은 10 이다. 이 값은 Karlin 과 Altschul 의 stochastic model 에 의하면 10 개의 서열의 일치는 우연히 일어날 수 있다고 정의한다..

S, CUTOFF: high-scoring segment pair 들의 cutoff 를 정하는 옵션으로 기본값은 EXPECT value 로부터 계산 된 값을 이용한다.

MATRIX: 유사성 검색에 이용되는 scoring matrix 의 종류로 기본값은 BLOSUM62 이다. BLASTP, BLASTX, TBLASTN, TBLASTX 의 경우 옵션으로 PAM40, PAM120, PAM250, IDENTITY 등의 matrix 를 이용한다. 하지만 BLASTN 의 경우에는 다른 MATRIX 를 선택하실 수 없다.

STRAND: TBLASTN 의 경우에는 top 혹은 bottom strand 중 선택하여 검색을 할 수 있다. 그리고 BLASTN, BLASTX, TBLASTX 의 경우에도 quary sequence 중 top 혹은 bottom strand 의 open reading frame 을 선택하여 검색할 수 있다.

FILTER: 통계적으로는 중요한 값을 가지지만 생물학적으로는 의미가 없는 서열들을 제거하는 옵션이다. Low compositional complexity 를 가진 서열들은 Wootton 과 Federhen 에 의해 개발된 SEG program 이 이용되고 internal repeat 들은 Claverie 와 States 에 의해 개발된 XNU program 을 이용한다. BLASTN 의 경우에는 Tatusov 와 Lipman 에 의해 개발된 DUST 가 이용된다. 입력한 서열 중 일부가 low complexity sequence 로 인식이 되면 blast 는 염기의 경우 "N"으로 단백질의 경우 "X"로 표시한다. 그래서 실제 입력을 정확히 했음에도 불구하고 결과와 함께 출력되는 입력 서열에는 "NNNNNNN"혹은 "XXXXXXX"가 포함되어 있는 것을 가끔 볼 수 있다. 기본값은 filtering 을 하는 program 들을 이용하게 되어있고 사용자가 원하는 경우 filter 를 선택하지 않을 수 있다. Filter 의 기능은 오직 입력한 서열에 한해서만 적용된다.

NCBI-gi: accession number 와 locus name 이외에도 gi 번호를 같이 보여줄 수 있는 옵션이다.

마. BLAST의 결과 해석

기본적으로 결과의 형태는 FASTA의 결과 출력 형태와 유사하다. 결과는 P-value 순으로 보여준다. 일반적으로 가장 큰 의미를 가지는 값은 P-value와 high score이다. 의미가 있다고 생각되는 단백질 서열의 경우 P-value는 가능한 한 작아야 하고 high score는 커야 한다. DNA의 경우 의미있는 서열의 P-value가 0.0001 보다 작더라도 두 서열이 연관에 없을 가능성이 크다.

High score 보다는 P-value가 서열의 길이에 대해 영향을 받지 않으므로 homologous 한 서열을 판단할 때 중요한 기준이 된다. P-value가 e^{-100}보다 작은 경우는 일반적으로 같은 종의 같은 서열로 고려된다. P-value가 e^{-50}-e^{-100}사이일 때는 아주 유사한 서열로 고려할 수 있다. P-value가 e^{-10}-e^{-50}사이일 때는 연관된 서열로 고려할 수 있다. P-value가 0.1-e^{-5}사이일 때는 연관성을 가질 가능성은 있으나 상당히 먼 관계를 가지고 있을 가능성이 있다. 일반적으로 P-value가 0.1 보다 큰 경우에는 큰 의미를 가진다고 할 수 없다.

바. BLAST 2.0

BLAST 2.0은 기존의 blast에 gap을 도입하는 기능이 추가되었다. 일반적으로 blast 검색을 수행하면 결과가 끊어진 몇 개의 조각들로 출력이 되는데, blast 2.0에서는 gap을 도입하여 FASTA와 같이 insertion과 deletion을 도입하여 상동성이 있는 조각들을 연결하여 결과를 보여준다 (Alschul et.al.,1997). 그 이외의 옵션은 기본적인 blast의 옵션과 동일하고 다른 부분은 다음과 같다.

- Graphical Overview : 결과에서 입력한 서열과 유사한 부분을 그림으로 표시해 주는 옵션이다.
- Query Genetic Codes (blastx only) : blastx에서 translation을 할 때 어떤 genetic code를 사용하는 가를 선택할 수 있는 옵션이다. BLAST2.0 검색 결과로는 'bit' score와 Expect value를 보여준다. 사용한 scoring matrix에 따라 값이 달라지는 high score와는 달리 bit score는 scoring matrix의 영향을 받지 않는다. Expect value는 통계적인 의미가 있으며 특정 데이터베이스에서 우연히 (by chance) 해당 점수를 가지면서 배열될 수 있는 서열들의 갯수를 의미한다. 또 두 서열 사이의 expect value는 우연히 해당 점수를 가지며 배열하는 확률을 의미한다. 따라서 값이 작을 수록 서열은 더 의미가 있으며 검색 결과 서열들도 E-value가 작은 순서대로 나열한다. Expect value는 데이터베이스의 크기에 따라 바뀌며 (데이터베이스가 커질 0수록 우연히 해당 점수의 배열을 할 서열의 수를 많아짐) 결과를 배열하는데 기준으로 하기 좋은 값이며 같은 query로 다른 데이터베이스 검색 결과들을 비교할 때 사용할 수 있는 값이다.

사. PSI-BLAST

또한 NCBI에서는 PSI(Position-Specific Iterated)-BLAST를 최근 개발하여 서비스 하고 있다. PSI-BLAST는 일반 BLAST의 기능에 motif이나 profile의 비교 기능을 추가한 프로그램이다. 즉 PSI-BLAST는 기본적인 BLAST 검색을 수행한 후 그 결과를 이용하여 multiple alignment를 수행한다. Multiple alignment를 통해 position-specific score matrix를 제작하고 이 matrix를 이용하여 다시 BLAST 검색을 수행한다. 즉 일반 검색과 motif, profile 검색을 동시에 수행하게 되는 것이다. 진화적으로 멀리 떨어져 있는 homolog 서열을 찾는데 유리하게 사용될 수 있다.

3. 유사성 비교 프로그램들의 성능 비교

유사성 비교 프로그램들의 적절한 사용을 위해 몇 개의 bioinformatics group 에서 여러 조건에서 각 프로그램들의 성능 비교를 수행하였다. 성능 비교는 프로그램들이 임의의 서열을 입력하였을 때 진화적으로 유사한 서열들을 얼마나 잘 찾아내는가에 초점을 두고 있다. 그래서 대부분의 성능 비교는 분석용 데이터베이스를 만든 후 검색을 수행하여 true positive, false negative, true negative, false positive 들의 비율을 조사하는 방식을 많이 이용한다.

	진화적인 유사성	기준(threshold) 값과의 비교
True positive	있음	기준 값보다 높음
False negative	있음	기준 값보다 낮음
True negative	없음	기준 값보다 낮음
False positive	없음	기준 값보다 높음

(표 6) true positive, false negative, true negative, false positive 의 정의

또한 각 프로그램들의 직접적인 비교를 위해 equivalence number, minimum errors, the receiver operating characteristic 등의 세 가지 기준값을 사용한다.

(1) Equivalence Number: (EN)

False positive 와 false negative 의 개수가 같아지는 threshold score 를 equivalence score 로 정의하고, equivalence score 에서의 false positive 의 개수를 EN 으로 정의한다. 즉 EN 은 프로그램의 sensitivity (false negatives)와 selectivity (false positives)를 동시에 고려한 값이 된다.

(2) The Minimum Number of Errors (MER)

MER 은 EN 값의 2 배한 값을 상향 값으로 한다. 즉 2 X EN ≥ MER 로 나타내고 threshold score 를 true positive 와 true negative 사이의 값으로 정의한다. 예를 들면 100 개의 homolog 와 1000 개의 non-homolog 가 있다고 가정하고 유사성 검색을 했을 때 결과가 80 개의 homolog, 1 개의 non-homolog, 20 개의 homolog, 999 개의 non-homolog 의 순서로 출력되었다고 가정하자. 이경우 정의에 의해 EN 과 MER 은 모두 1 의 값을 가진다. 하지만 EN 의 threshold 는 1 개의 non-homolog 와 20 개의 homolog 사이에 위치하게 되는 반면, MER 의 threshold 는 20 개의 homolog 와 999 개의 non-homolg 사이의 값을 가지게 된다. 따라서 MER 의 threshold 가 보다 생물학적으로 의미있는 값을 가지게 된다.

(3) The receiver operating characteristic (ROC)

ROC 는 앞의 두 비교 값 보다 positive hit 과 negative hit 의 순서에 중점을 둔 값이다. ROC curve 는 다음과 같이 정의할 수 있는 selectivity(P−)에 대한 sensitivity (P+)의 함수로 나타내어 지는 그래프이다.

Sensitivity: $P^+ = t^+ / (t^+ + f^-)$
Selectivity: $P^- = t^- / (t^- + f^+)$

ROC (performance measure)는 ROC curve 의 밑 부분영역의 numerical integration 을 통해 구한 값을 의미한다. ROC 값은 유사성 검색 프로그램의 검색의 정확도를 측정하는 중요한 값이 된다. 즉 완벽한 검색 프로그램은 true

negative 와 false positive 가 없음을 의미하므로 1 의 값을 가지고, 가장 좋지 않은 프로그램은 반대로 0 의 값을 가진다. 일반적으로 ROC 는 0 과 1 사이의 값을 가지게 된다.

3.1 단백질 서열 분석

최근 개발된 유사성 검색 프로그램들에 대한 성능비교는 Agarwal 등에 의해 수행되었다 (Agarwal and States, 1998). 이들이 검색에 사용한 프로그램들은 SSEARCH (Release 3.0t74), FASTA (Release 3.0t74), Probabilistic Smith-Waterman (PSW), BLASTP (Release 1.4.9MP), WU-BLAST2 (Release 2.0a1MP) 이며 67 개의 알려진 protein family 를 이용하여 검색을 수행하였다. PSW 는 최근 Bucher 와 Hoffman 에 의해 개발되었으며, 기존의 dynamic programming 과 Hidden Morkov model(HMM)을 결합한 검색 프로그램이다. 검색은 다음과 같이 각 검색 프로그램들의 기본 조건에서 수행하였다.

	Scoring matrix	Gap penalty	Ranking order
SSEARCH	BLOSUM 50	-12, -2	z-score
FASTA	BLOSUM 50	-12, -2	z-score
PSW	BLOSUM 45	-8, -4	
BLASTP	BLOSUM 62	Not permited	Sum statistics
WU-BLAST2	BLOSUM 62	-9, -2	Sum statistics of gap penalties

(표 7) 서열 분석 프로그램들의 성능 비교 조건

표 8 은 검색 결과의 MER 값을 나타낸 표이다. 표에서 Length 는 입력 서열의 길이를, Size 는 protein family 에 속한 서열들의 개수를, PS 는 PSW 를, SSEARCH 는 Smith-Waterman 을, FA 는 FASTA 를, BP 는 BLASTP 를, B2 는 WU-BLAST2 를 의미한다. 마지막 column 은 가장 작은 MER 값을 가지는 검색 프로그램, 즉 제일 좋은 수행 결과를 출력하는 프로그램들의 이름을 적었다. 비교에 사용된 모든 프로그램들이 비슷한 출력을 수행하였을 때는 빈칸으로 남겨 놓았다. 마지막 두 라인은 모든 family 의 검색에서 도출된 false positive 와 false negative 의 개수를 기록하였다.

Description/superfamily	Length	Size	PS	SW	FA	BP	B2	Best method
L-Lactate dehydrogenase	333	26	0	0	2	3	0	PS/SW/B2
E2 protein papillomavirus	322	26	0	0	0	0	0	
Core antigen-hepatitis B	183	25	0	0	0	0	0	
Antithrombin-III	464	25	0	0	1	0	0	PS/SW/BP/B2
Thymidine kinase	376	25	1	1	1	0	1	BP
Phycocyanin	162	25	0	0	0	0	0	
Protamine Y2	34	24	2	1	1	3	1	SW/FA/B2
Transforming prot. (myc)	439	24	0	0	0	0	0	

Matrix protein	348	24	0	0	6	0	0	PS/SW/BP/B2
H+-transporting A TPsynthase P6	226	23	1	1	8	4	1	PS/SW/B2
Alcohol dehydrogenase A	375	23	0	0	0	0	0	
Glycoprotein B	857	23	0	0	0	0	0	
Ionotropic acetylcholine receptor	457	23	0	0	0	0	0	
Non-structural protein NS2	121	22	0	1	1	2	2	PS
Annexin I	346	22	4	4	4	4	4	
Histone H1b	218	22	3	2	3	2	2	SW/BP/B2
Metallothionein	61	21	6	6	6	6	3	B2
[beta]-Crystallin chain Bp	204	21	0	0	0	0	0	
Proteinase inhibitor	71	21	2	3	3	3	3	PS
Hepatic lectin H1	291	20	2	1	1	7	1	SW/FA/B2
E2 glycoprotein precursor	1447	20	0	0	0	0	0	
[alpha]-2u-Globulin precursor	181	20	6	9	10	10	9	PS
Pepsin	388	20	0	0	0	0	0	
DNA-directed DNA polymerase	1462	20	5	1	1	1	2	SW/FA/BP
Prolactin	227	20	0	0	0	0	0	
Vitamin B12 trans. btuD	249	20	3	7	9	6	10	PS
Total		3544	355	367	490	510	394	
False positives			109	126	110	159	102	
False negatives			246	241	380	351	292	

(표 8) 단백질 검색 프로그램들의 성능 비교

표 8 에서 protein family 에 따라 약간식 다른 결과가 나왔지만 대부분의 경우 PSW 와 SW 가 좋은 수행 결과를 나타냄을 알 수 있다.

일반적으로 검색 결과는 입력 서열의 길이에 의존하는 것으로 알려져 있으므로 Agarwal 등은 검색 길이에 따른 검색 수행 결과를 계산하였다. 표 9 에서 e0 와 e1 은 full sequence 를, e0-b62 에서 11 까지의 set 은 같은 검색 조건에서 길이를 점점 짧게 한 partial sequence 를 나타낸다. (+)는 각 프로그램들이 잘 구별한 protein family 의 개수를 보여주고, (+=)는 pairwise comparision 을 통해 잘 구별하거나 비슷하게 구별한 protein family 의 개수를 보여준다. 각 프로그램들중 가장 좋은 검색을 수행한 프로그램들을 이탤릭 체로 표시하였다. ROC 는 50 개의 negative 가 발견되는 곳을 기준으로 하였다.

Method		PSW		SW		FASTA		BLASTP		BLASTP2	
Query set	Stat	+	+ =	+	+ =	+	+ =	+	+ =	+	+ =
e0	MER	*10*	41	4	*47*	1	34	2	32	3	42
e0	ROC	*10*	34	4	*35*	7	25	4	26	7	32
e0	EN	*11*	39	1	*46*	2	34	2	29	3	43
e1	MER	*8*	41	3	*43*	1	35	6	35	2	42
e1	ROC	*13*	*31*	4	*31*	6	27	6	26	5	*31*
e1	EN	*6*	44	2	*48*	2	38	4	36	2	45
e0-b62	MER	*12*	33	2	*43*	2	32	2	32	3	42
e0-b62	ROC	*10*	29	7	*33*	5	26	4	26	5	33
e0-b62	EN	*11*	33	1	*44*	3	32	2	30	3	*44*
l0	MER	3	23	2	*37*	3	36	*11*	34	3	35
l0	ROC	8	11	8	*21*	6	18	*12*	17	*12*	18
l0	EN	3	25	6	33	3	34	*8*	32	3	*39*
l1	MER	5	24	4	32	5	31	*13*	32	4	*35*
l1	RO	10	15	8	21	7	21	*13*	21	6	*23*
l1	EN	*8*	23	2	37	5	34	*8*	33	2	*40*

(표 9) 길이에 따른 단백질 서열 분석 프로그램의 성능 비교

표 8 에서와 마찬가지로 full length protein sequence 의 검색에서는 PSW 와 SW 가 가장 좋은 결과를 산출함을 알 수 있으나 길이가 짧아짐에 따라 이러한 특성은 없어지는 것을 볼 수 있다. 오히려 짧은 서열의 경우에는 BLASTP 가 가장 좋은 결과를 산출하는 것을 알 수 있다.

3.2 유전자 서열 분석

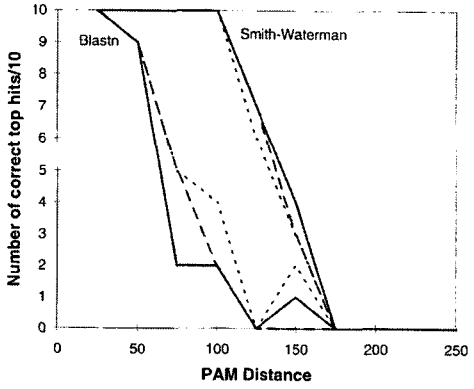
2 장에서 기술한 단백질 서열 분석의 유리함에도 불구하고 다음과 같은 이유에서 유전자 서열 분석 또한 중요한 의미를 가진다.

(1) 대부분의 경우 유전자 서열이 데이터베이스에 입력된 후 annotation 과정을 거쳐 다시 단백질 데이터베이스에 자료가 입력되므로 유전자 데이터베이스가 가장 최신의 서열정보를 포함하고 있다.

(2) 입력된 유전자 서열의 잘못된 translation 에 의해 단백질 데이터베이스에 틀린 서열 정보가 입력 될 수 있다. 즉 유사성이 있는 염기 서열이 데이터베이스에 저장되어 있음에도 불구하고 단백질을 이용한 유사성 검색시 잘못된 translation 에 의해 유사성을 찾지 못하는 경우가 발생 할 수 있다.

(3) 입력한 서열이 non-coding region 이거나 정확한 open-reading frame 을 찾기 힘든 EST 일 경우 단백질 서열의 비교를 통해서는 분석이 힘들다.

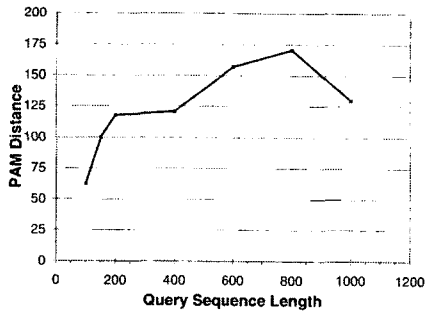
Anderson 과 Brass 는 효과적인 유전자 서열 분석을 위해 유사성 검색 프로그램들의 성능을 조사하였다 (Anderson and Brass, 1998). Andersion 과 Brass 는 우선 서열의 길이가 다른 10 개의 서열을 선발하고 이것들을 PAM distance 를 변화시키면서 인위적으로 진화 시킨 1000 개의 서열을 만든 후 검색을 수행하였다. 각 PAM distance 에 따른 BLASTN 과 Smith-Waterman 의 비교 결과는 다음 그래프와 같다. 그래프에서 실선은 ciliary neurotrophic factor 를, - - - - 은 tyrosine phosphatase 를, ... 은 retinol binding protein 을 나타낸다.



(그림 4) PAM distance 에 따른 Blastn 과 Smith-Waterman 의 비교

그림 4 에서 출력 결과는 단백질 서열의 종류에 관계없이 유사한 pattern 으로 나옴을 알 수 있고, full length 유전자 서열을 입력하였을 경우 Smith-Waterman 이 좀 더 좋은 결과를 출력함을 알 수 있다. 즉 BLASTN 의 경우 PAM distance 가 50 이상이 되면 유사성 검색 결과를 검색 할 수 있는 능력이 급격히 떨어지는데 비해 Smith-Waterman 의 경우는 PAM distance 가 100 이상이 될 때까지 정확하게 homologous protein 들을 검색 하는 것을 알 수 있다.

또한 Anderson 과 Brass 는 Smith-Waterman 의 길이에 따른 검색 결과를 조사 하였으며 (50 %의 homologous family 를 검색하는 것을 기준) 그 결과는 다음과 같다.



(그림 5) 서열 길이에 따른 Smith-Waterman 의 selectivity 의 변화

그림 5 의 경우 단백질 서열 검색에서와 마찬가지로 검색 서열의 길이가 짧아질 경우 Smith-Waterman 은 full-length 서열 검색에서와 같이 좋은 검색 결과를 내지 못함을 알 수 있다. 또한 Anderson 과 Brass 는 ROC 값의 조사를 통해 염기 서열의 길이가 200 bp 이상일 경우 각 염기 서열 분석 프로그램에서 다음과 같은 cut-off value 를 제시 하였다. 즉 임의의 서열을 비교하였을 때 통계값이 다음에서 제시하는 값 범위 이내에 들 경우 검색된 서열은 입력한 서열과 homologous 할 가능성이 크다는 것을 의미한다.

BLASTN (default)	p value: • 0.01
BLASTN (w=6)	p value: • 0.01
BLAST2 (default)	p value: • 0.005
FASTA	E value: • 0.005
SW	Z-score: • 5

4. 맺음말

서열의 유사성 검색은 새로이 밝혀낸 서열의 특성 파악과 기능 예측을 위해 가장 먼저 시도 하는 분석 방법이다. 하지만 유사성 검색 프로그램들의 결과로 출력된 서열들 중 의미가 있는 것과 그렇지 못한 것을 구별하는 것은 쉽지 않다. 본 review 에서는 생물학자들이 각 프로그램들을 보다 효율적으로 사용하기 위해 현재 가장 많이 이용되고 있는 세가지의 분석 프로그램 (Smith-Waterman, FASTA, BLAST)의 분석 원리를 간략히 설명하고, 각 프로그램들의 비교 분석 결과를 제시하였다.

일반적인 유사성 검색에서 유전자 서열의 비교 보다는 단백질 서열의 비교가 보다 정확한 유사성 검색을 수행 할 수 있다. 하지만 가장 최근에 밝혀진 서열의 검색이나 translation 이 잘못된 단백질 서열에 의해 생기는 error 를 막기 위해서는 유전자 서열을 통한 유사성 검색도 같이 진행하는 것이 바람직하다

Full length 서열의 경우 유전자 서열과 단백질 서열에 관계 없이 Smith-Waterman 방법이 가장 좋은 유사성 비교를 수행하고 EST 같은 partial sequence 에서는 BLAST 가 오히려 더 좋은 유사성 비교를 수행한다. 하지만 기술한 유사성 분석 프로그램들은 단지 서열의 유사성 만을 비교할 뿐 서열들이 가지고 있는 특징들 (진화적으로 보존된 부분 혹은 3 차원 구조등) 에 대해서 고려하지는 않는다. 그러므로 새로운 서열이 밝혀진 경우 기본적인 유사성 검색을 통해 homologous 할 가능성이 많은 서열들이 찾아진 경우 서열의 특징들을 이용하여 분석하는 2 차 분석 도구들 (PROSITE, MOTIF, PRINTS 등) 을 이용하여 확인하는 것이 바람직하다.

5. 참고 문헌

Agarwal, P. and States, D.J. *Bioinformatics*, 14,1, 40 (1998)

Altschul, S.F., Gish, W., Miller, W. Myers, M. and Lipman, D.J. *J. Mol. Biol.* 215,403 (1990)

Altschul, S.F., Madden, T.L., Schaffer, A.A., Zhang, J., Miller, W. and Lipman, D.J. *Nucleic Acids Research 25*, 3389 (1997)

Anderson, I. and Brass, A. *bioinformatics 14*, 349. (1998)

Dayhoff, M.O., Schwartz, R.M. and Orcutt, B.C. A model of evolutionary change. In Dayhoff, M.O. (ed), *Atlas of protein sequence and structure*. National Biomedical Research Foundation, Washington, DC, Volume 5 Suppl.3, pp.345-352(1978)

Pearson, W.R. and Lipman. D.J. *Proc. Natl. Acad.Sci.U.S.A.* 85, 2444 (1988)

Smith, T.F. and Waterman, M.S. *J. Mol.Biol.* 147, 195 (1981)