LETTER

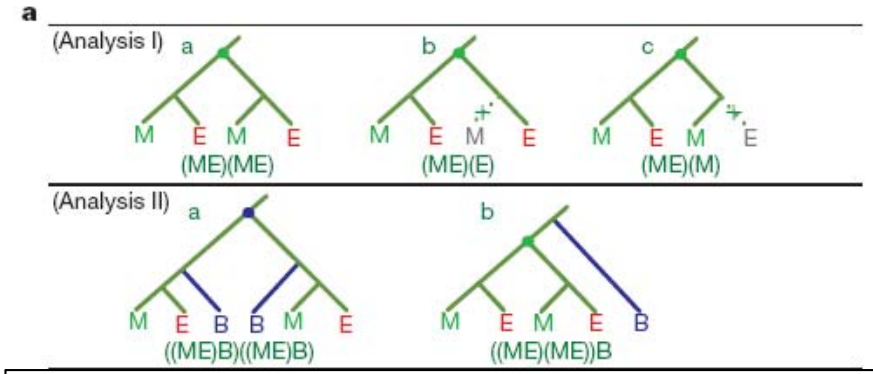
doi:10.1038/nature09916

Ancestral polyploidy in seed plants and angiosperms

Yuannian Jiao[1,2], Norman J. Wickett[2], Saravanaraj Ayyampalayam[3], André S. Chanderbali[4], Lena Landherr[2], Paula E. Ralph[2], Lynn P. Tomsho[5], Yi Hu[2], Haiying Liang[6], Pamela S. Soltis[7], Douglas E. Soltis[4], Sandra W. Clifton[8], Scott E. Schlarbaum[9], Stephan C. Schuster[5], Hong Ma[1,2,10,11], Jim Leebens-Mack[3] & Claude W. dePamphilis[1,2]

Whole-genome duplication (WGD), or polyploidy, followed by gene loss and diploidization has long been recognized as an important evolutionary force in animals, fungi and other organisms[1–3], especially plants. The success of angiosperms has been attributed, in part, to innovations associated with gene or whole-genome duplications[4–6], but evidence for proposed ancient genome duplications pre-dating the divergence of monocots and eudicots remains equivocal in analyses of conserved gene order. Here we use comprehensive phylogenomic analyses of sequenced plant genomes and more than 12.6 million new expressed-sequence-tag sequences from phylogenetically pivotal lineages to elucidate two groups of ancient gene duplications—one in the common ancestor of extant seed plants and the other in the common ancestor of extant angiosperms. Gene duplication events were intensely concentrated around 319 and 192 million years ago, implicating two WGDs in ancestral lineages shortly before the diversification of extant seed plants and extant angiosperms, respectively. Significantly, these ancestral WGDs resulted in the diversification of regulatory genes important to seed and flower development, suggesting that they were involved in major innovations that ultimately contributed to the rise and eventual dominance of seed plants and angiosperms.

Angiosperms are by far the largest group of land plants, with more than 300,000 living species. Significantly, most flowering plant lineages reflect one or more rounds of ancient polyploidy. For example, extensive analyses of the complete genome sequence of *Arabidopsis thaliana*



최근 피자식물들 중 가장 먼저 분지된 그룹(가장 원시적인)인 *Amborella*의 유전체 전체가 밝혀졌다. 이를 이용하여 분석해 본 결과 기존에 알려져 있던 피자식물의 진화 이후 *Arabidopsis* 까지의 세 번의 WGD 이외에 이끼류에서 *Amborella*까지의 진화 동안에도 두 번의 WGD 가 더 일어났음을 밝혔다. 이로서 WGD현상은 관속식물의 진화과정 중 빈번히 일어나서 관속식물의 다양화 현상을 일으킨 기본적인 기작 일이 밝혀졌다.

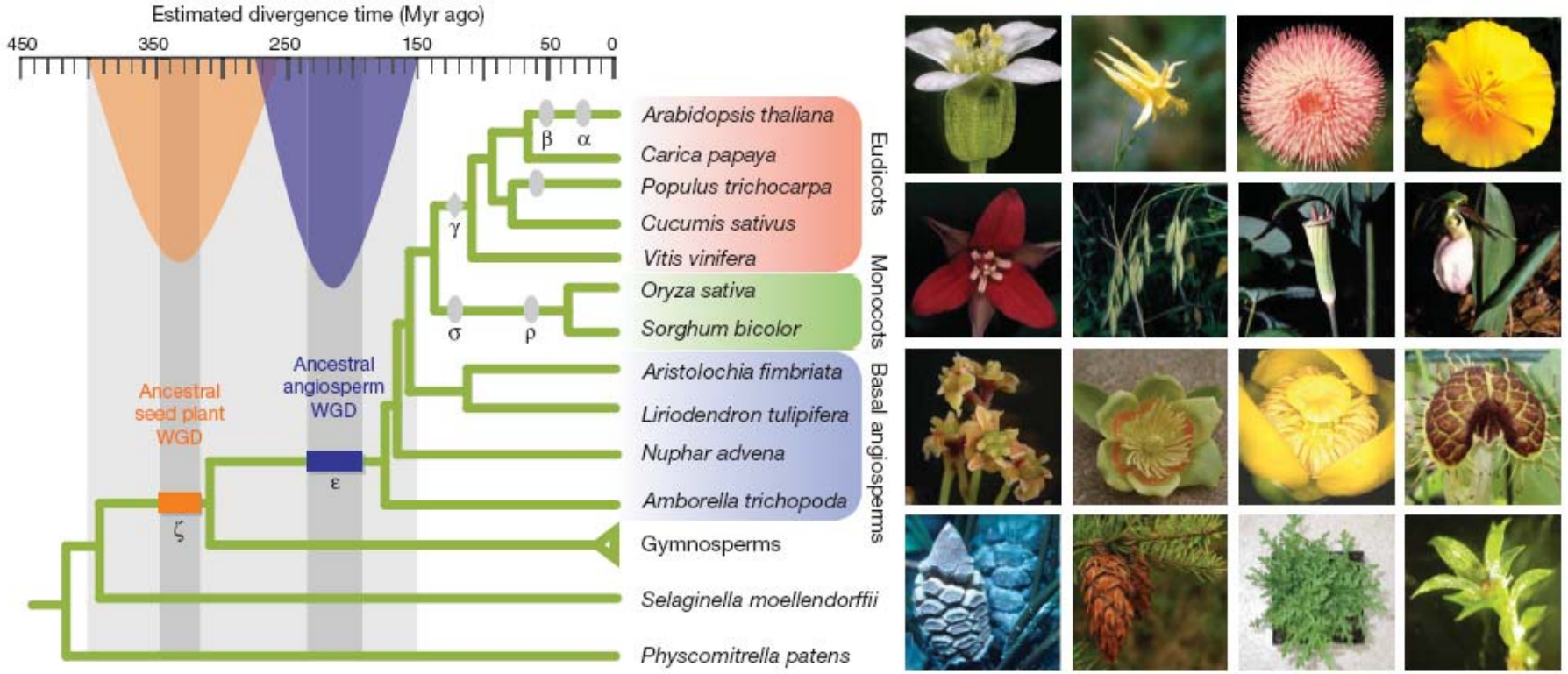


Figure 3 | Ancestral polyploidy events in seed plants and angiosperms. Two ancestral duplications identified by integration of phylogenomic evidence and molecular time clock for land plant evolution. Ovals indicate the generally accepted genome duplications identified in sequenced genomes (see text). The diamond refers to the triplication event probably shared by all core eudicots. Horizontal bars denote confidence regions for ancestral seed plant WGD and ancestral angiosperm WGD, and are drawn to reflect upper and lower bounds of mean estimates from Fig. 2 (more orthogroups) and Supplementary Fig. 5 (more taxa). The photographs provide examples of the reproductive diversity of eudicots (top row, left to right: *Arabidopsis thaliana*, *Aquilegia chrysantha*, *Cirsium pumilum*, *Eschscholzia californica*), monocots (second row, left to right: *Trillium erectum*, *Bromus kalmii*, *Arisaema triphyllum*, *Cypripedium acaule*), basal angiosperms (third row, left to right: *Amborella trichopoda*, *Liriodendron tulipifera*, *Nuphar advena*, *Aristolochia fimbriata*), gymnosperms (fourth row, first and second from left: *Zamia vazquezii*, *Pseudotsuga menziesii*) and the outgroups *Selaginella moellendorfii* (vegetative; fourth row, third from left) and *Physcomitrella patens* (fourth row, right). See Supplementary Table 4 for photo credits.

http://news.sciencemag.org/sciencenow/2011/04/double-the-genes-double-the-flor.html?ref=hp

아래 예를 다시 계산해 봅시다.

Example 2
Alignment between ACACT and ACT.
(match score =1, mismatch score=0, gap penalty=-1)

Sequence 1 ←

Sequence 2 →

		A	C	A	C	T
	0	-1	-2	-3	-4	-5
A	-1					
C	-2					
T	-3					

Gap penalty

1. Upper value + gap penalty
2. Left value + gap penalty
3. Upper left value + match/mismatch score

아래 예를 다시 계산해 봅시다.

Example 2
Alignment between ACACT and ACT.
(match score =1, mismatch score=0, gap penalty=-1)

Sequence 1 ←

Sequence 2 →

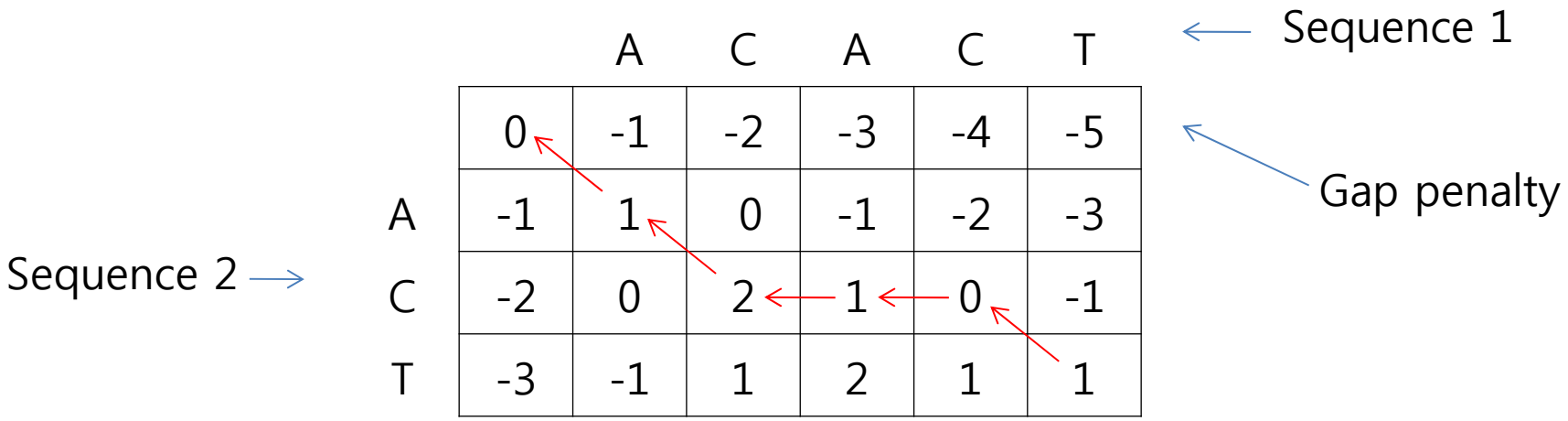
		A	C	A	C	T
	0	-1	-2	-3	-4	-5
A	-1	1	0	-1	-2	-3
C	-2	0	2	1	0	-1
T	-3	-1	1	2	1	1

Gap penalty

1. Upper value + gap penalty
2. Left value + gap penalty
3. Upper left value + match/mismatch score

아래 예를 다시 계산해 봅시다.

Example 2
Alignment between ACACT and ACT.
(match score =1, mismatch score=0, gap penalty=-1)

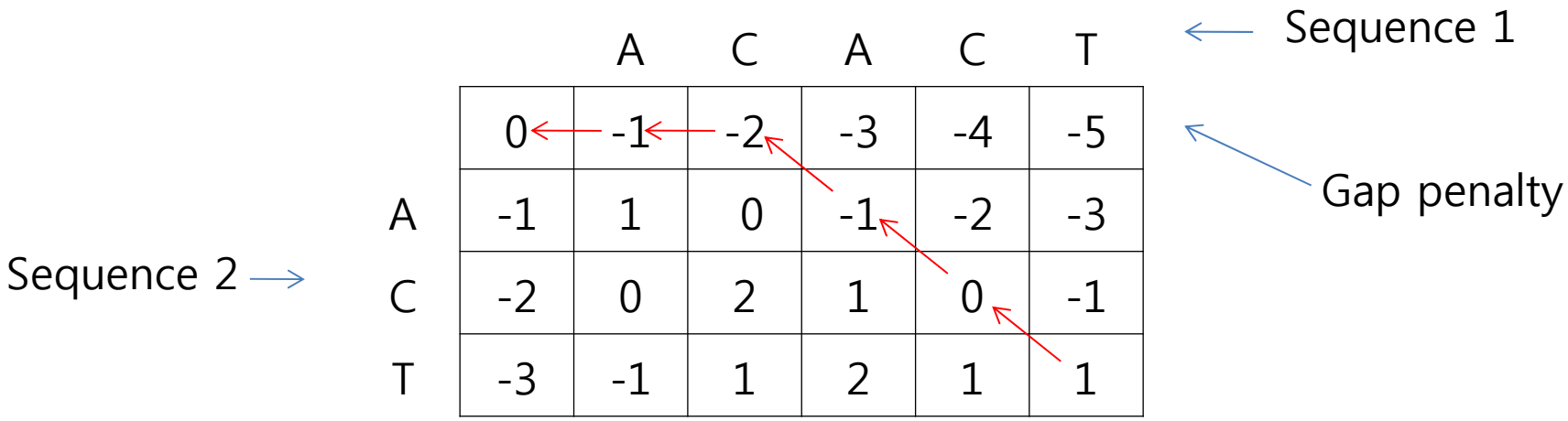


1. Upper value + gap penalty
2. Left value + gap penalty
3. Upper left value + match/mismatch

ACACT
AC--T

아래 예를 다시 계산해 봅시다.

Example 2
Alignment between ACACT and ACT.
(match score =1, mismatch score=0, gap penalty=-1)



1. Upper value + gap penalty
2. Left value + gap penalty
3. Upper left value + match/mismatch

ACACT
--ACT

두 가지의 경로가 나옴.
→ 이 경우 terminal gap과 internal gap을 만드는 2가지 경우가 생김

Semiglobal alignment 반전역정렬

- 긴 염기서열과 짧은 염기서열을 정렬할 때
- Internal gap과 terminal gap을 다르게 점수를 주는 것.

  → 왜냐하면 아마도 terminal gap이 생긴 것은 길이가 다르기 때문일 수 있기 때문.
- Semiglobal alignment를 찾기 위해서 기본적인 basic dynamic programming algorithm에서 다음과 같은 두 가지를 변형을 함.

  1. initial gap에 대한 penalty를 주지 않기 위해 table의 첫번째 줄과 열을 모두 "0"으로 놓는다.

  2. end gap에 대한 penalty를 주지 않기 위해 마지막 column과 마지막 row에서의 이동 시 no gap penalty

Semiglobal alignment 반전역정렬

match score =1, mismatch score=0, gap penalty=-1

1. Table의 첫째 열과 첫째 열을 모두 "0"로...

   첫째 행에서 수평이동 하는 것은 좌측 서열에 initial gap을 추가하는 것임.

   → 좌측 서열에는 initial gap에 대한 penalty가 없다.

   첫째 열에서 수직이동 하는 것은 위의 서열에 initial gap을 추가하는 것임.

   → 위의 서열에는 initial gap에 대한 penalty가 없다.

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

Semiglobal alignment 반전역정렬

2. 마지막 열의 수직이동과 마지막 행의 수평이동은 no gap penalty

- 맨 마지막 cell의 경우 (gap penalty = -1):
  Vertical move: 0 ← NO GAP PENALTY
  Horizontal move: 3 ←
  Diagonal move: 0 + match score = 1

  → maximum score of the 3 movements = 3

		A	A	C	A	G	T	C	T
	0	0	0	0	0	0	0	0	0
A	0	1	1	0	1	0	0	0	0
G	0	0	1	1	0	2	1	0	0
T	0	0	0	1	1	1	3	3	3

Semiglobal alignment 반전역정렬

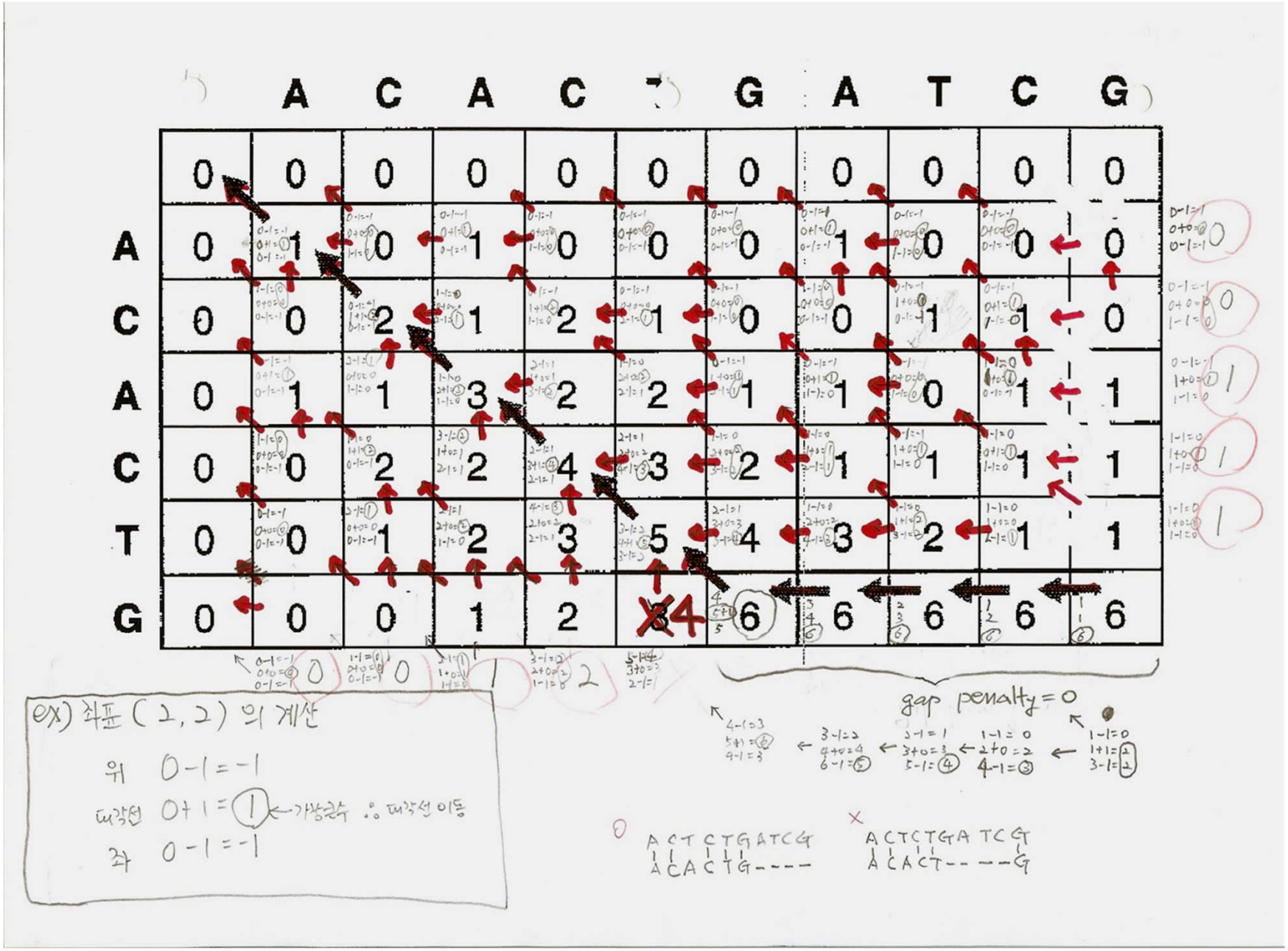
2. 마지막 열의 수직이동과 마지막 행의 수평이동은 no gap penalty

- 맨 마지막 cell의 경우 (gap penalty = -1):
  Vertical move: 0 ⟵ NO GAP PENALTY
  Horizontal move: 3 ↙
  Diagonal move: 0 + match score = 1

→ maximum score of the 3 movements = 3

		A	A	C	A	G	T	C	T
	0 ←	0 ←	0 ←	0	0	0	0	0	0
A	0	1	1	0	1	0	0	0	0
G	0	0	1	1	0	2	1	0	0
T	0	0	0	1	1	1	3 ←	3 ←	(3)

AACAGTCT
---AGT--


A C A C T G A T C G
A 0 1 0 1 0 0 1 0 0 0
C 0 0 2 1 2 1 0 0 1 0
A 0 1 1 3 2 2 1 1 0 1
C 0 0 2 2 4 3 2 1 1 1
T 0 0 1 2 3 5 4 3 2 1
G 0 0 0 1 2 4 6 6 6 6
gap penalty = 0
ex) 좌표 (2, 2) 의 계산
위 0-1=-1
대각선 0+1=1 ← 가장큰수 ∴ 대각선 이동
좌 0-1=-1
ACTCTGATCG
ACACTG----
ACTCTGATCG
ACACT---G

Local alignment 국부정렬

- 보다 flexible한 방법
- 두 sequence간의 가장 잘 일치하는 "부분서열"을 찾아내는것.

- **Smith-Waterman algorithm**:
  - basic dynamic programming algorithm 에서 쓰는 3종류의 movement (vertical, horizontal, and diagonal) 대신
    → negative score는 무조건 zero 로 처리

  - Table의 첫째 줄과 열은 모두 zero

  - 모든 score들을 기입한 후 최대값을 갖는 score를 찾아 이 곳에서부터 점수가 0에 이를 때 까지 사선으로 이동

  - 사선이 그어진 부분이 best local alignment임.

Local alignment 지역정렬

- Example:
  - Find the local alignment for the following sequences: (match score=1, mismatch score=0, gap penalty=-1)

ATTCGATCC
ACGAT

- initial the partial score table as the following

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

Local alignment 지역정렬

- Fill in each of partial score with one of the four options
  (Score of vertical move, horizontal move, diagonal move or zero)

		A	T	T	C	G	A	T	C	C
	0	0	0	0	0	0	0	0	0	0
A	0	1	0	0	0	0	1	0	0	0
C	0	0	1	0	1	0	0	1	1	1
G	0	0	0	1	0	2	1	0	1	1
A	0	1	0	0	1	1	3	2	1	0
T	0	0	2	1	0	1	2	4	3	2

Local alignment 지역정렬

- Find the maximum partial score in the table
- Work backward until reach a zero

		A	T	T	C	G	A	T	C	C
	0	0	0	0	0	0	0	0	0	0
A	0	1	0	0	0	0	1	0	0	0
C	0	0	1	0	1	0	0	1	1	1
G	0	0	0	1	0	2	1	0	1	1
A	0	1	0	0	1	1	3	2	1	0
T	0	0	2	1	0	1	2	4	3	2

Maximum partial score

- Best local alignment:

CGAT
CGAT

● Smith-Waterman algorithm에 의한 best local alignment를 찾아내는 것이 실제 가장 많이 쓰임.
→ BLAST search!

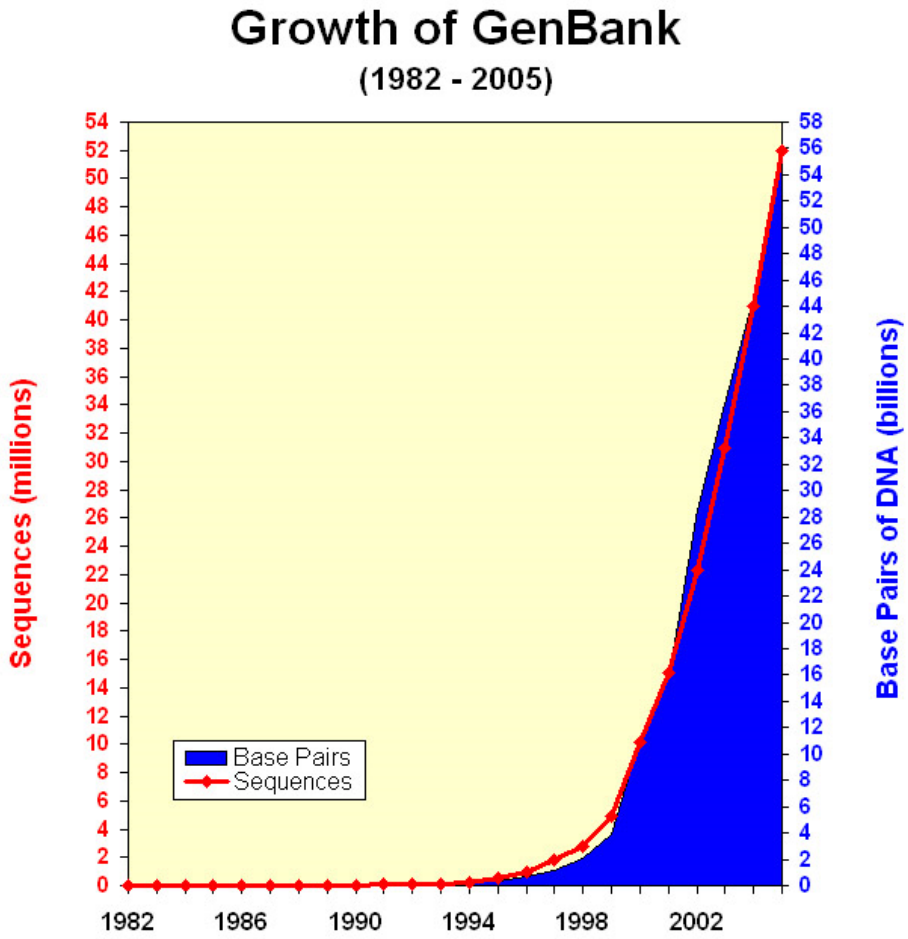
● 어떤 경우에 databank search를 하나?

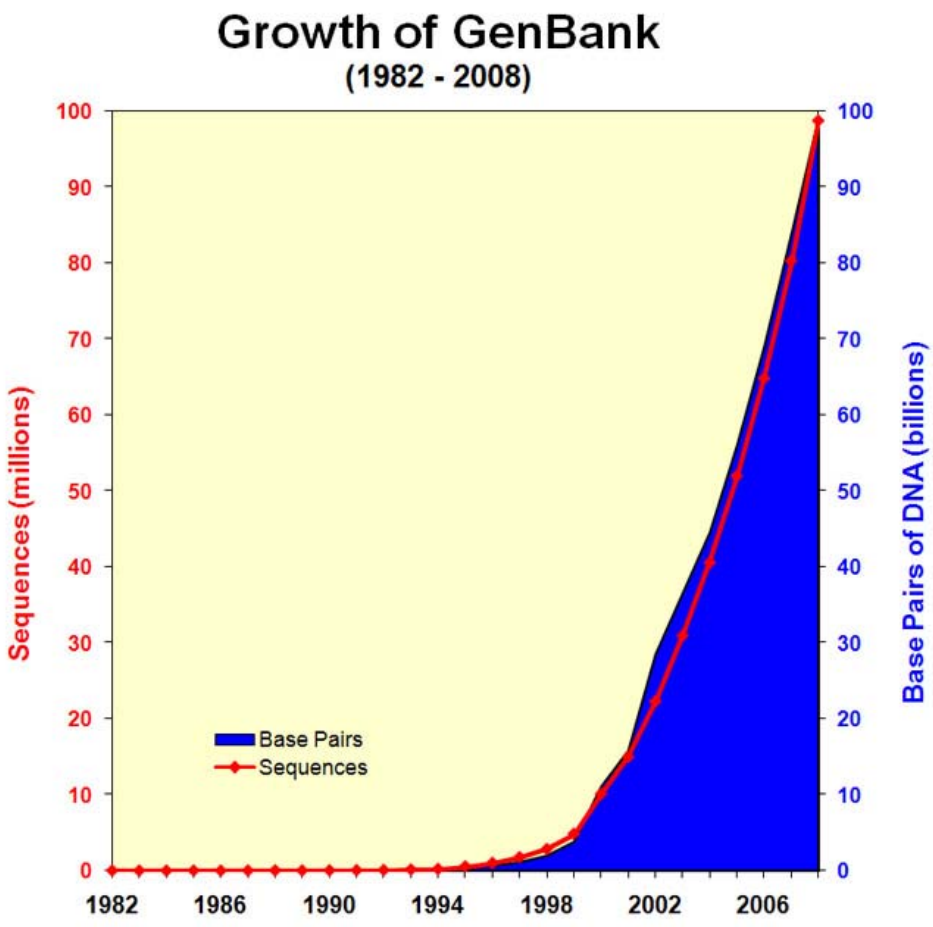
- new/unknown genes (proteins) 에 대한 동정

- unknown genes (protein)에 대한 기능 유추

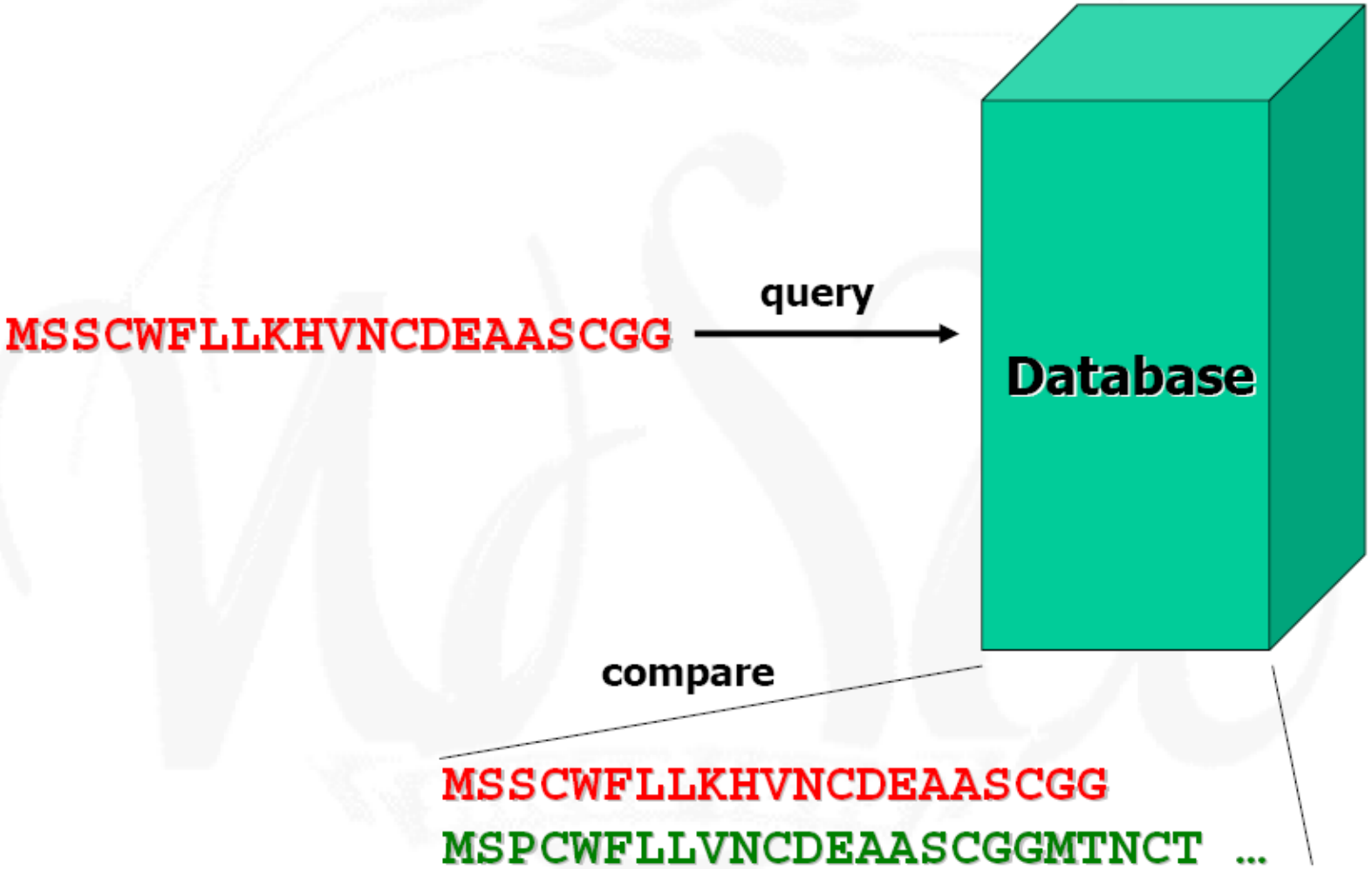
- 다른 관련 연구들을 위한 관련 sequence 수집:

phylogenetic analyses
primer design
looking for new motifs

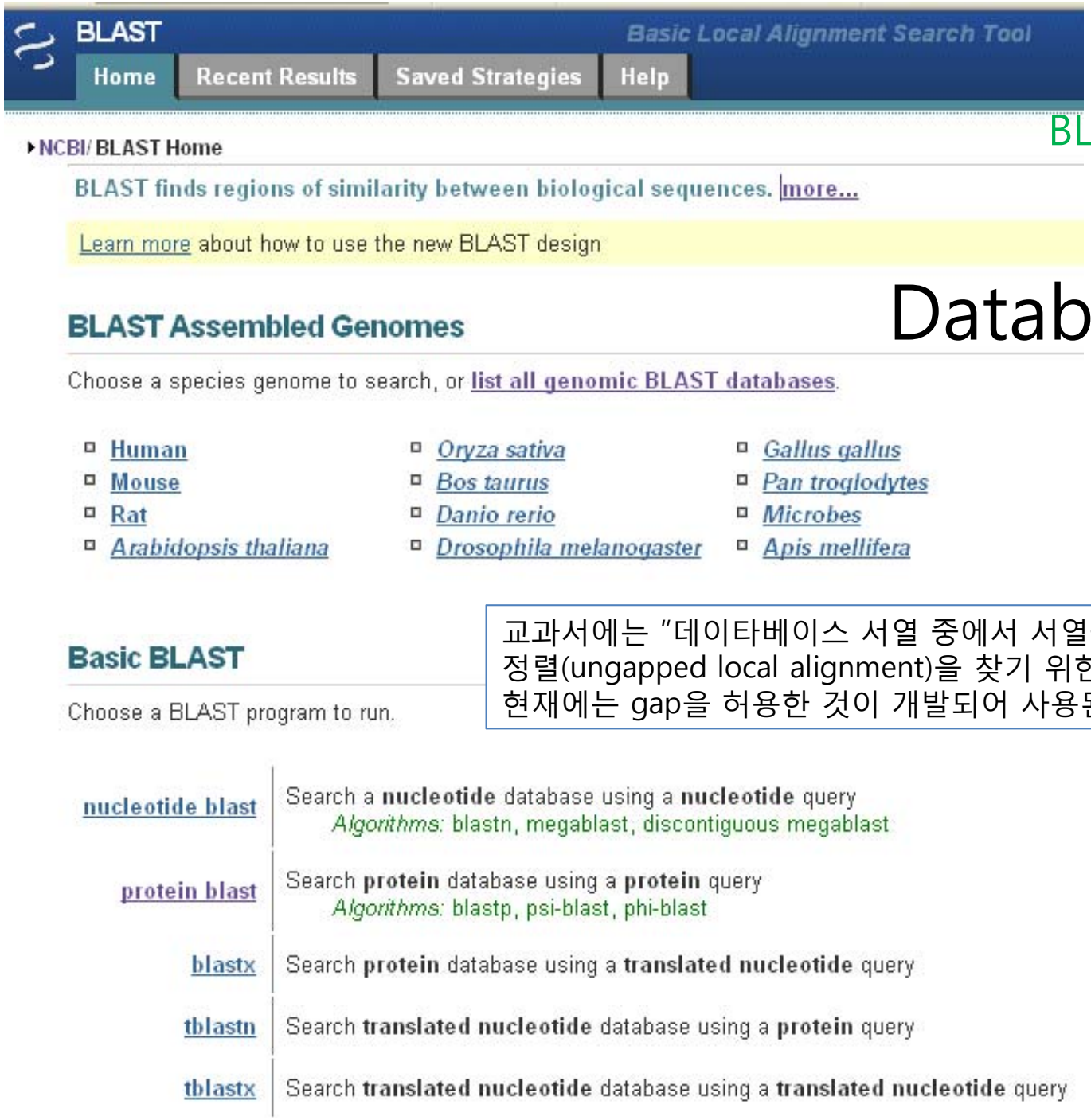

Growth of GenBank
(1982 - 2005)
Sequences (millions)
Base Pairs of DNA (billions)
Base Pairs
Sequences
1982
1986
1990
1994
1998
2002


Growth of GenBank
(1982 - 2008)
Sequences (millions)
Base Pairs of DNA (billions)
Base Pairs
Sequences
1982
1986
1990
1994
1998
2002
2006

Basics


MSSCWFLLKHVNCDEAASCGG
query
Database
compare
MSSCWFLLKHVNCDEAASCGG
MSPCWFLLVNCDEAASCGGMTNCT ...

Database Search

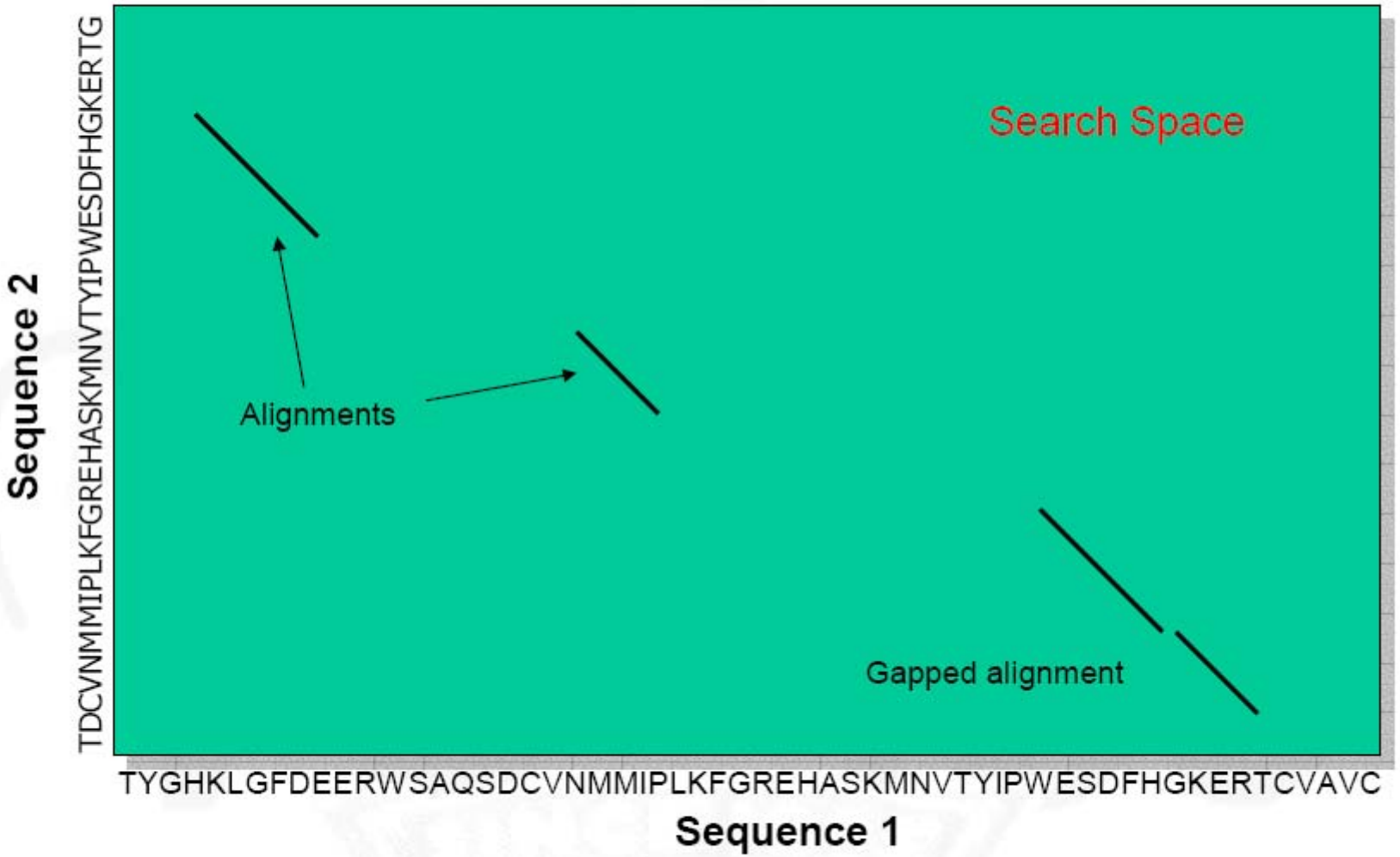


교과서에는 "데이타베이스 서열 중에서 서열공백이 허용되지 않은 국부 정렬(ungapped local alignment)을 찾기 위한 것이다"라고 되어 있지만 현재에는 gap을 허용한 것이 개발되어 사용된다.

BLAST and Its Relatives

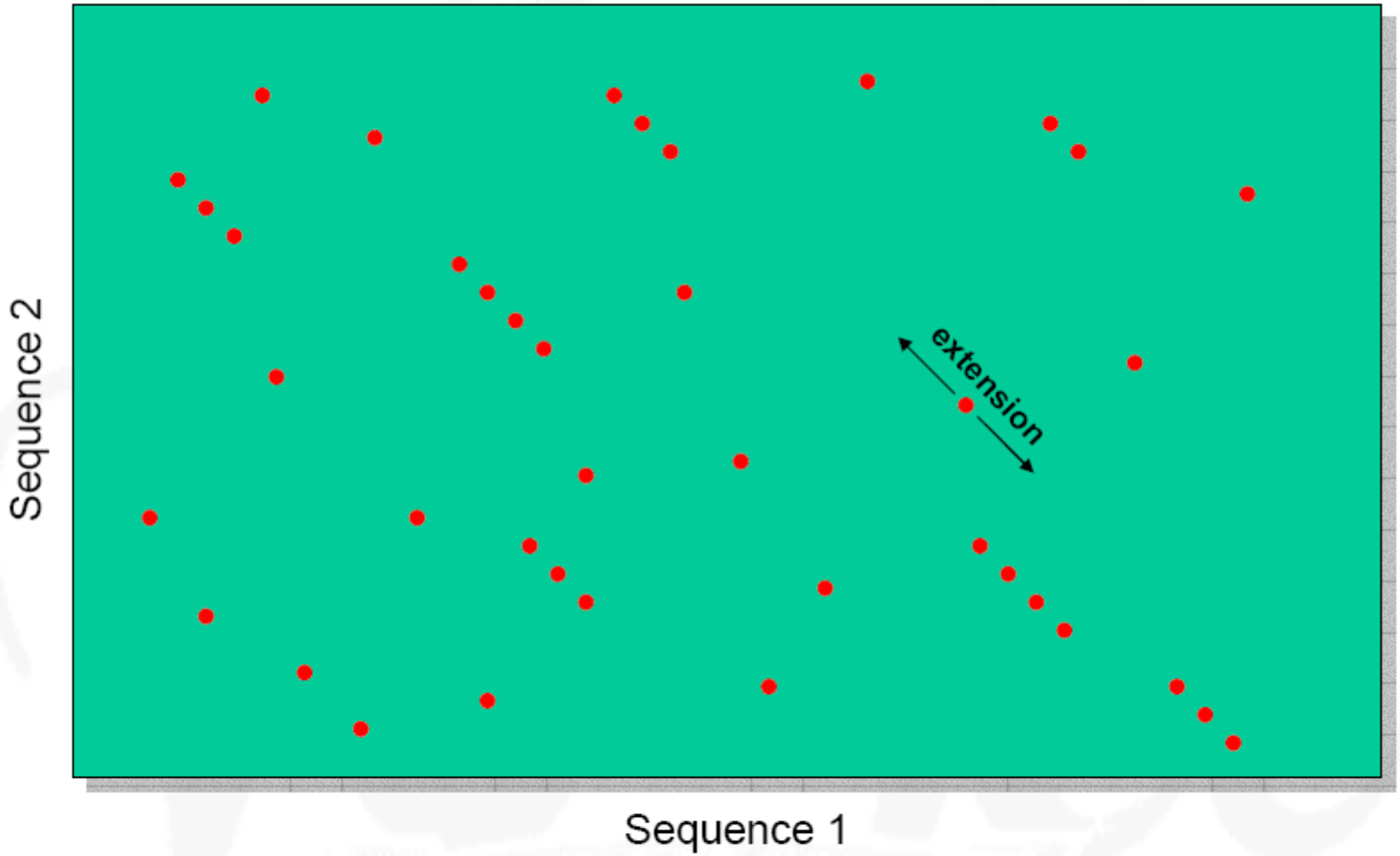
Program	Database	Query	Typical uses
BLASTN	Nucleotide	Nucleotide	Mapping oligonucleotide, cDNAs, and PCR products to a genome, etc.
BLASTP	Protein	Protein	Identifying common regions between proteins; collecting related proteins for phylogenetic analyses
BLASTX	Protein	Nucleotide translated into protein	Finding protein-coding genes in genomic DNA; determining if a cDNA corresponds to a known protein
TBLASTN	Nucleotide translated into protein	Protein	Identifying transcripts similar to a given protein; mapping a protein to genomic DNA
TBLASTX	Nucleotide translated into protein	Nucleotide translated into protein	Cross species gene prediction at the genome or transcript level, etc.

BLAST Algorithm



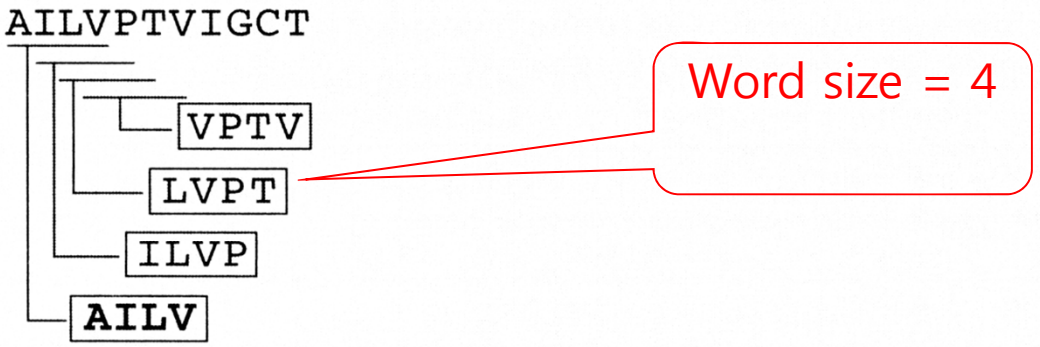
1. Seeding **2. Extension** **3. Evaluation**

Extension



Input sequence: AILVPTV

1) Break the query sequence into words



2) Search for word matches (also called high-scoring pairs, or HSPs) in the database sequences

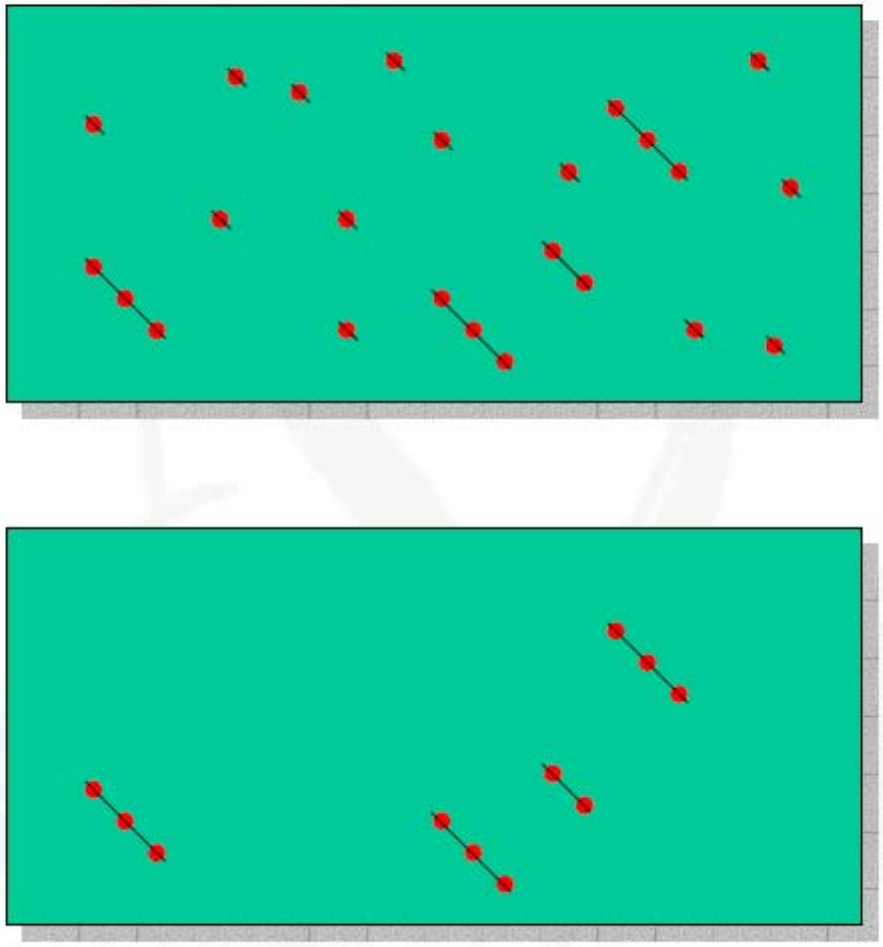
AILV
MVQGWALYDFLKCR**AILV**GTVIAML . . .

3) Extend the match until the local alignment score falls below a fixed threshold (the most recent version of BLAST allows gaps in the extended match)

AILVPTVI
MVQGWALYDFLKCR**AILVGTVI**AML . . .

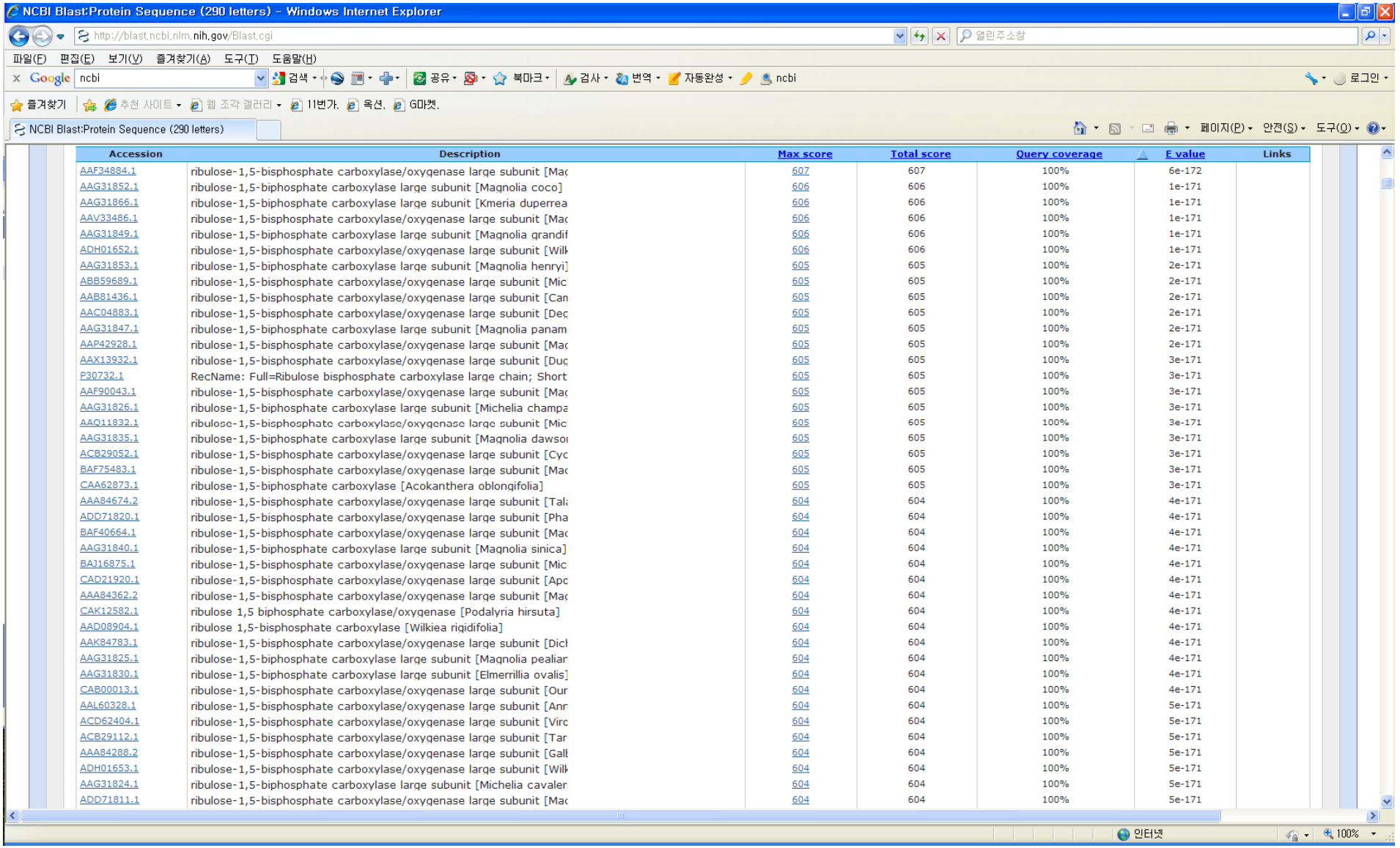
FIGURE 2.11 *Overview of the BLASTP search process.*

Evaluation



High-scoring pairs (HSPs)

score threshold

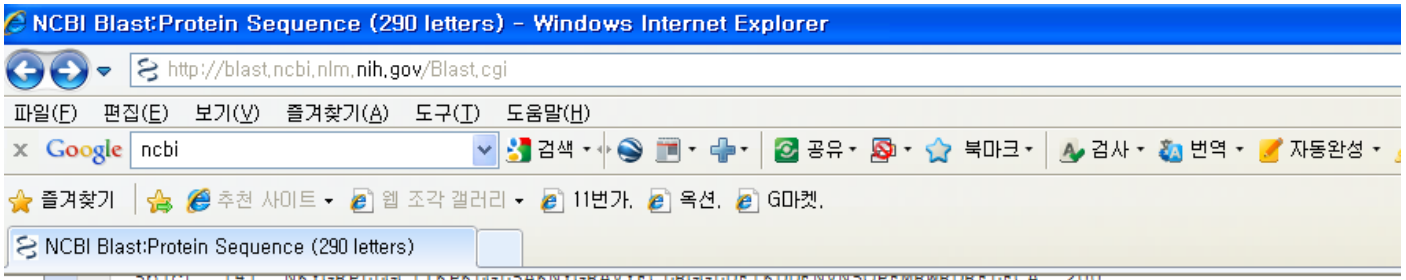


Accession	Description	Max score	Total score	Query coverage	E value	Links
AAF34884.1	ribulose-1,5-bisphosphate carboxylase/oxygenase large subunit [Mac	607	607	100%	6e-172	
AAG31852.1	ribulose-1,5-biphosphate carboxylase large subunit [Magnolia coco]	606	606	100%	1e-171	
AAG31866.1	ribulose-1,5-biphosphate carboxylase large subunit [Kmeria duperrea	606	606	100%	1e-171	
AAV33486.1	ribulose-1,5-biphosphate carboxylase/oxygenase large subunit [Mac	606	606	100%	1e-171	
AAG31849.1	ribulose-1,5-biphosphate carboxylase large subunit [Magnolia grandif	606	606	100%	1e-171	
ADH01652.1	ribulose-1,5-bisphosphate carboxylase/oxygenase large subunit [Wilk	606	606	100%	1e-171	
AAG31853.1	ribulose-1,5-biphosphate carboxylase large subunit [Magnolia henryi]	605	605	100%	2e-171	
ABB59689.1	ribulose-1,5-bisphosphate carboxylase/oxygenase large subunit [Mic	605	605	100%	2e-171	
AAB81436.1	ribulose-1,5-bisphosphate carboxylase/oxygenase large subunit [Can	605	605	100%	2e-171	
AAC04883.1	ribulose-1,5-bisphosphate carboxylase/oxygenase large subunit [Dec	605	605	100%	2e-171	
AAG31847.1	ribulose-1,5-biphosphate carboxylase large subunit [Magnolia panam	605	605	100%	2e-171	
AAP42928.1	ribulose-1,5-bisphosphate carboxylase/oxygenase large subunit [Mac	605	605	100%	2e-171	
AAX13932.1	ribulose-1,5-bisphosphate carboxylase/oxygenase large subunit [Duc	605	605	100%	3e-171	
P30732.1	RecName: Full=Ribulose bisphosphate carboxylase large chain; Short	605	605	100%	3e-171	
AAF90043.1	ribulose-1,5-bisphosphate carboxylase/oxygenase large subunit [Mac	605	605	100%	3e-171	
AAG31826.1	ribulose-1,5-biphosphate carboxylase large subunit [Michelia champa	605	605	100%	3e-171	
AAQ11832.1	ribulose-1,5-bisphosphate carboxylase/oxygenase large subunit [Mic	605	605	100%	3e-171	
AAG31835.1	ribulose-1,5-biphosphate carboxylase large subunit [Magnolia dawsoi	605	605	100%	3e-171	
ACB29052.1	ribulose-1,5-bisphosphate carboxylase/oxygenase large subunit [Cyc	605	605	100%	3e-171	
BAF75483.1	ribulose-1,5-bisphosphate carboxylase/oxygenase large subunit [Mac	605	605	100%	3e-171	
CAA62873.1	ribulose-1,5-biphosphate carboxylase [Acokanthera oblongifolia]	605	605	100%	3e-171	
AAA84674.2	ribulose-1,5-bisphosphate carboxylase/oxygenase large subunit [Tal	604	604	100%	4e-171	
ADD71820.1	ribulose-1,5-bisphosphate carboxylase/oxygenase large subunit [Pha	604	604	100%	4e-171	
BAF40664.1	ribulose-1,5-bisphosphate carboxylase/oxygenase large subunit [Mac	604	604	100%	4e-171	
AAG31840.1	ribulose-1,5-biphosphate carboxylase large subunit [Magnolia sinica]	604	604	100%	4e-171	
BAJ16875.1	ribulose-1,5-bisphosphate carboxylase/oxygenase large subunit [Mic	604	604	100%	4e-171	
CAD21920.1	ribulose-1,5-bisphosphate carboxylase/oxygenase large subunit [Apc	604	604	100%	4e-171	
AAA84362.2	ribulose-1,5-bisphosphate carboxylase/oxygenase large subunit [Mac	604	604	100%	4e-171	
CAK12582.1	ribulose 1,5 biphosphate carboxylase/oxygenase [Podalyria hirsuta]	604	604	100%	4e-171	
AAD08904.1	ribulose 1,5-bisphosphate carboxylase [Wilkiea rigidifolia]	604	604	100%	4e-171	
AAK84783.1	ribulose-1,5-bisphosphate carboxylase/oxygenase large subunit [Dicl	604	604	100%	4e-171	
AAG31825.1	ribulose-1,5-biphosphate carboxylase large subunit [Magnolia pealiar	604	604	100%	4e-171	
AAG31830.1	ribulose-1,5-biphosphate carboxylase large subunit [Elmerrillia ovalis]	604	604	100%	4e-171	
CAB00013.1	ribulose-1,5-bisphosphate carboxylase/oxygenase large subunit [Our	604	604	100%	4e-171	
AAL60328.1	ribulose-1,5-bisphosphate carboxylase/oxygenase large subunit [Anr	604	604	100%	5e-171	
ACD62404.1	ribulose-1,5-bisphosphate carboxylase/oxygenase large subunit [Virc	604	604	100%	5e-171	
ACB29112.1	ribulose-1,5-bisphosphate carboxylase/oxygenase large subunit [Tar	604	604	100%	5e-171	
AAA84288.2	ribulose-1,5-bisphosphate carboxylase/oxygenase large subunit [Gall	604	604	100%	5e-171	
ADH01653.1	ribulose-1,5-bisphosphate carboxylase/oxygenase large subunit [Wilk	604	604	100%	5e-171	
AAG31824.1	ribulose-1,5-biphosphate carboxylase large subunit [Michelia cavaler	604	604	100%	5e-171	
ADD71811.1	ribulose-1,5-bisphosphate carboxylase/oxygenase large subunit [Mac	604	604	100%	5e-171	

- 데이타베이스에서 찾아진 sequence와 query sequence 간의 alignment score 를 S라 할 때

- E score: 데이타베이스 검색에서 몇 개의 서열이 random한 확률로 S 이상의 정렬점수를 갖는 것으로 발견되는가를 나타내는 기대치.

- P score: alignment score가 S 이상인 서열이 한 개 이상 무작위로 발견될 확률.




```
Sbjct  141  NKIGRPLLGCTIKPKLGLSAKNYGRAVYECLRGGLDFTKDDENVNSQPFMRWRDRFLFCA  200

Query  121  EAIYKAQAETGEIKGHYLNATAGTCEEMMKRAIFARELGVPIVMHDYLTGGFTANTTLAH  180
            EAIYK+QAETGEIKGHYLNATAGTCEEMMKRAIFARELGVPIVMHDYLTGGFTANT+LAH
Sbjct  201  EAIYKSQAETGEIKGHYLNATAGTCEEMMKRAIFARELGVPIVMHDYLTGGFTANTSLAH  260

Query  181  YCRDNGLLLHIHRAMHAVIDRQKNHGMHFRVLAKALRMSGGDHIHAGTVVGKLEGERDIT  240
            YCRDNGLLLHIHRAMHAVIDRQKNHGMHFRVLAKALRMSGGDHIHAGTVVGKLEGERDIT
Sbjct  261  YCRDNGLLLHIHRAMHAVIDRQKNHGMHFRVLAKALRMSGGDHIHAGTVVGKLEGERDIT  320

Query  241  LGFVDLLRDDFIEKDRSRGIYFTQDWVSLPGVLPVASGGIHVWHMPALTE  290
            LGFVDLLRDDFIEKDRSRGIYFTQDWVSLPGVLPVASGGIHVWHMPALTE
Sbjct  321  LGFVDLLRDDFIEKDRSRGIYFTQDWVSLPGVLPVASGGIHVWHMPALTE  370


>qb|ABG24979.1|  ribulose-1,5-bisphosphate carboxylase/oxygenase large subunit
[Echinopepon paniculatus]
Length=452

 Score =  602 bits (1553),  Expect = 1e-170, Method: Compositional matrix adjust.
 Identities = 287/290 (99%), Positives = 290/290 (100%), Gaps = 0/290 (0%)

Query  1    YPLDLFEEGSVTNMFTSIVGNVFGFKALRALRLEDLRIPTAYVKTFQGPPHGIQVERDKL  60
            YPLDLFEEGSVTNMFTSIVGNVFGFKALRALRLEDLRIPTAYVKTFQGPPHGIQVERDKL
Sbjct  80   YPLDLFEEGSVTNMFTSIVGNVFGFKALRALRLEDLRIPTAYVKTFQGPPHGIQVERDKL  139

Query  61   NKYGRPLLGCTIKPKLGLSAKNYGRAVYECLRGGLDFTKDDENVNSQPFMRWRDRFVFCA  120
            NKYGRPLLGCTIKPKLGLSAKNYGRAVYECLRGGLDFTKDDENVNSQPFMRWRDRF+FCA
Sbjct  140  NKYGRPLLGCTIKPKLGLSAKNYGRAVYECLRGGLDFTKDDENVNSQPFMRWRDRFLFCA  199

Query  121  EAIYKAQAETGEIKGHYLNATAGTCEEMMKRAIFARELGVPIVMHDYLTGGFTANTTLAH  180
            EAIYK+QAETGEIKGHYLNATAGTCEEMMKRAIFARELGVPIVMHDYLTGGFTANT+LAH
Sbjct  200  EAIYKSQAETGEIKGHYLNATAGTCEEMMKRAIFARELGVPIVMHDYLTGGFTANTSLAH  259

Query  181  YCRDNGLLLHIHRAMHAVIDRQKNHGMHFRVLAKALRMSGGDHIHAGTVVGKLEGERDIT  240
            YCRDNGLLLHIHRAMHAVIDRQKNHGMHFRVLAKALRMSGGDHIHAGTVVGKLEGERDIT
Sbjct  260  YCRDNGLLLHIHRAMHAVIDRQKNHGMHFRVLAKALRMSGGDHIHAGTVVGKLEGERDIT  319

Query  241  LGFVDLLRDDFIEKDRSRGIYFTQDWVSLPGVLPVASGGIHVWHMPALTE  290
            LGFVDLLRDDFIEKDRSRGIYFTQDWVSLPGVLPVASGGIHVWHMPALTE
Sbjct  320  LGFVDLLRDDFIEKDRSRGIYFTQDWVSLPGVLPVASGGIHVWHMPALTE  369
```

Select All Get selected sequences Distance tree of results Multiple alignment

FASTA algorithm

• BLAST와 더불어 일반적으로 많이 사용되는 searching algorithm and program. 이것의 입력 format을 다른 프로그램이나 데이타베이스에서 계속 사용하는 것임.

• database의 sequence들 중 query sequence와의 gap을 포함한 local alignment를 찾아내는 방법

• FASTA algorithm:

- 단백질의 경우 1-2개, nucleotide의 경우 4-6개로 구성된 "Word"로 나눔.
- 각각의 word의 위치를 보여주는 table을 만듬.
- 예를 들어 **MAEHGAHTFK**가 query sequence이면 (Word length=1)

Amino acid의 종류 →

word	A	C	D	E	F	G	H	I	K	L	M	N	P	Q	R	S	T	V	W	Y
pos	2			3	9	5	4		10		1						8			
	6						7													

FASTA algorithm

query sequence: **MAEHGAHTFK**

word	A	C	D	E	F	G	H	I	K	L	M	N	P	Q	R	S	T	V	W	Y
pos	2			3	9	5	4		10		1						8			
	6						7													

Query sequence 와 target sequence의 아미노산의 상대적 위치를 비교하여 보여주는 테이블을 하나더 만듬.
예: Target sequence가 **GALMNTEHMATH** 라고 하면

5-1=4 →

1	2	3	4	5	6	7	8	9	10	11	12
G	A	L	M	N	T	E	H	M	A	H	T
4	0		-3		2	-4	-4	-8	-8	-7	-4
	4								-4	-4	

FASTA algorithm

1	2	3	4	5	6	7	8	9	10	11	12
G	A	L	M	N	T	E	H	M	A	H	T
4	0		-3		2	-4	-4	-8	-8	-7	-4
	4								-4	-4	

• 이 테이블에서 가장 많은 같은 숫자를 보이는 것은 (-4)가 5회.
→ 가장 합리적인 정렬은 target sequence를 -4칸 이동시키는 것.

• 두 sequence들 간의 유사성은 위의 table에서 볼 수 있다.

• 이들 subsequence들은 합쳐져 큰 sequence들을 만든 후 Smith-Waterman algorithm을 사용하여 align 한다.

• FASTA는 유사한 구역을 이미 알고 있기 때문에 dynamic programming method보다 빠르다.