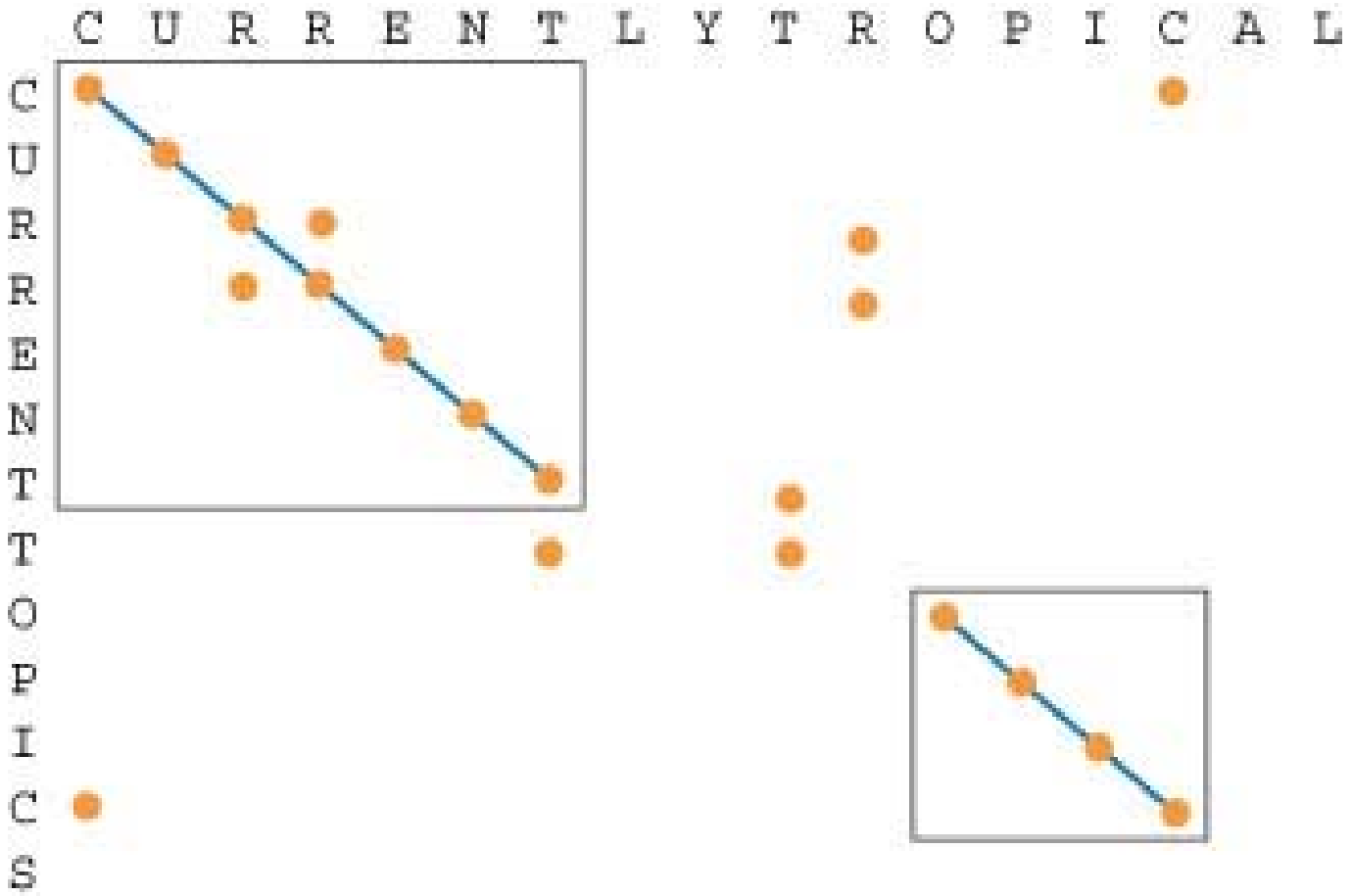
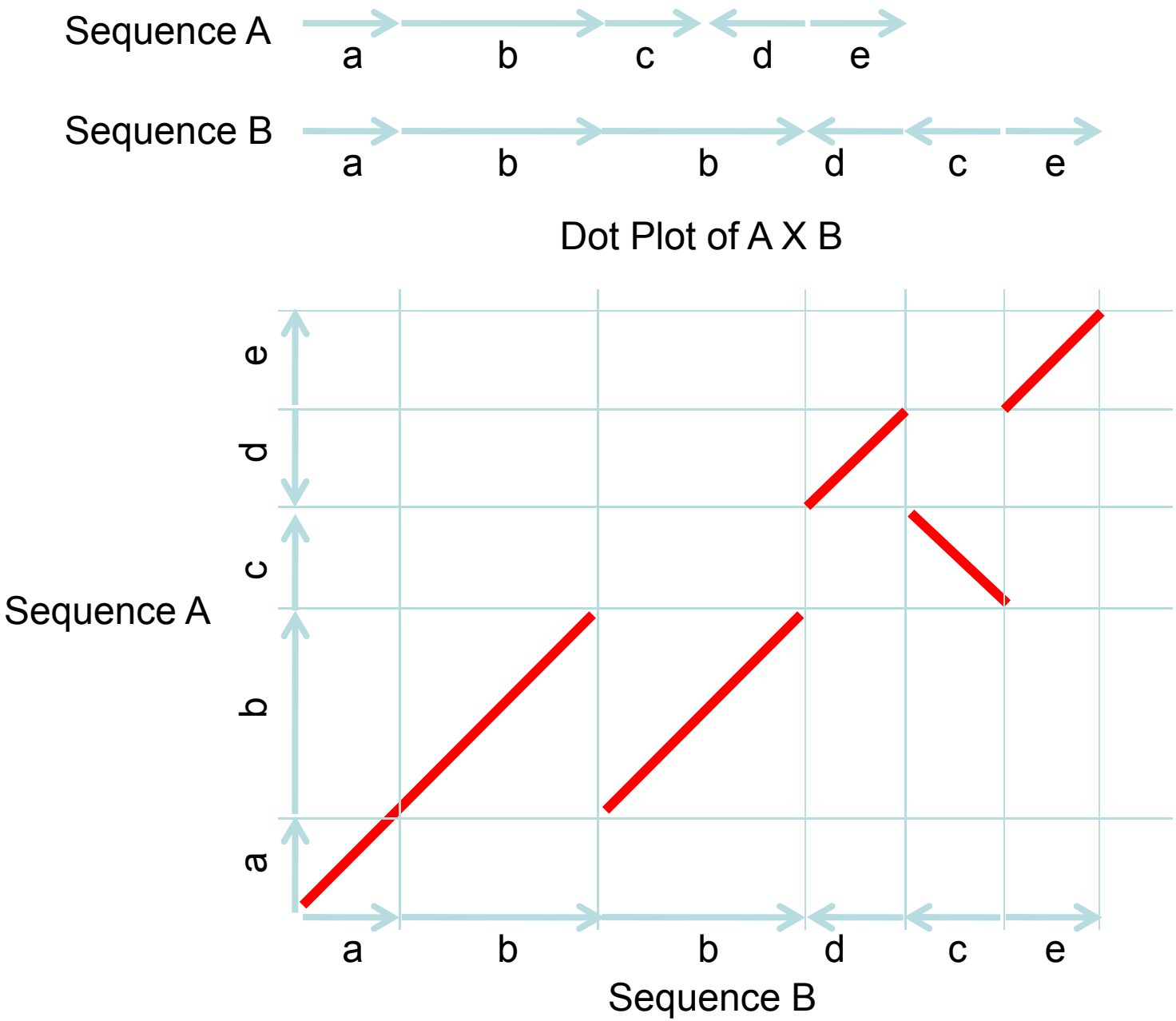
Chap. 2. Data Searches and Pairwise Alignment

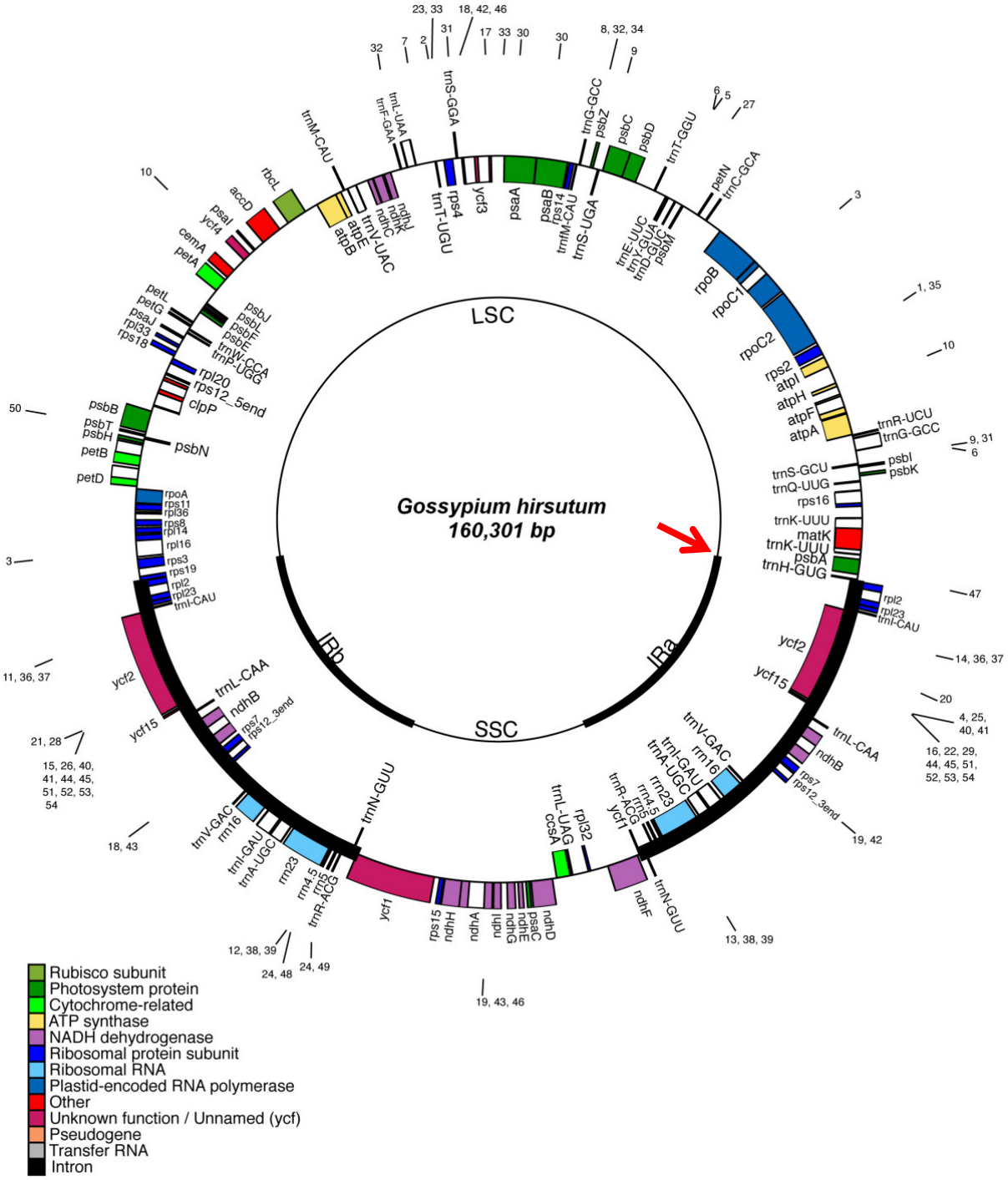
Dot plot



http://www.vivo.colostate.edu/molkit/dnadot/

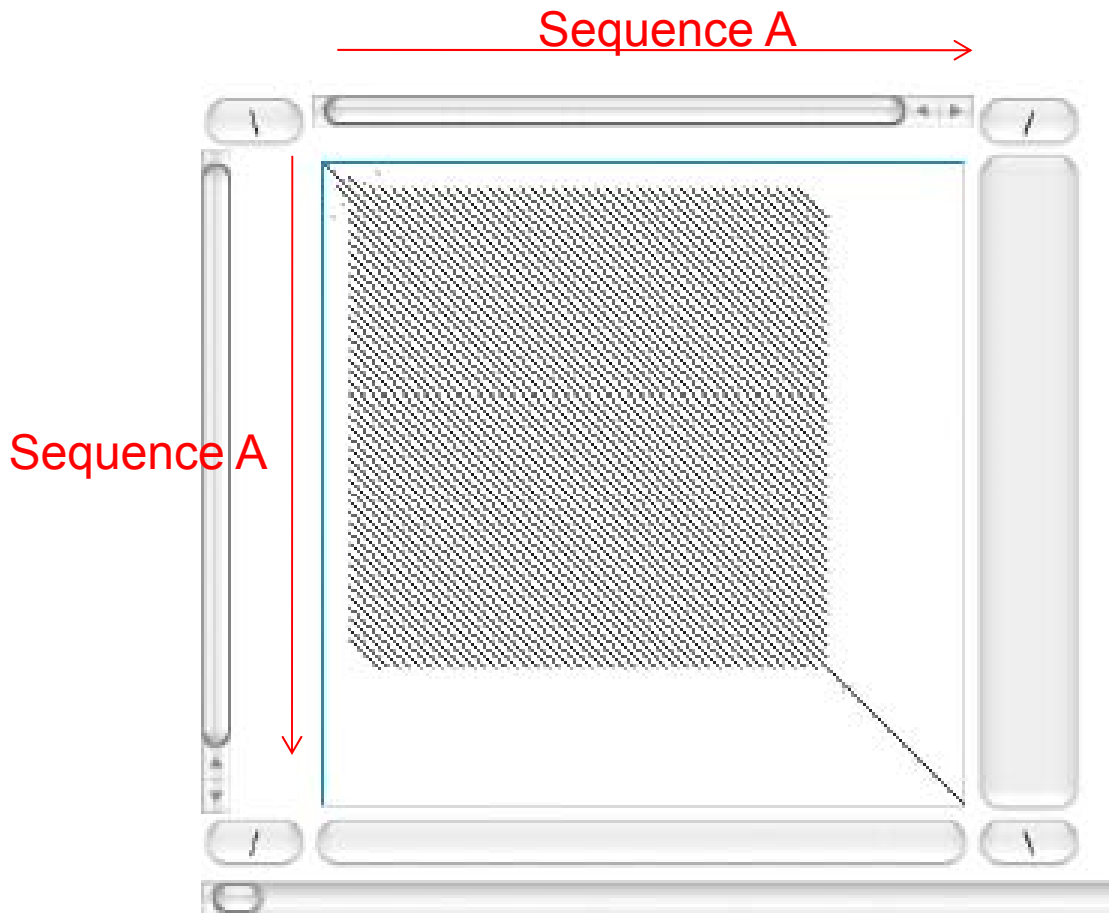
dot plot은 가로세로 축에 염기서열 또는 아미노산 서열을 배열하고 서로 같은 부분에 점을 찍어 나타낸 것이다. 특정 염기서열 또는 아미노산 서열의 내부 구조를 파악하기 위해서 dot plot을 사용한다.
주로 긴 서열은 일정단위 window 구간으로 잘라 서로 비교한다(예를 들어 100bp 구간으로 잘라 95%이상 같으면 동일한 것으로 취급하여 점을 찍음).


Sequence A
a
b
c
d
e
Sequence B
a
b
b
d
c
e
Dot Plot of A X B
Sequence A
e
d
c
b
a
a
b
b
d
c
e
Sequence B



Q: 오른쪽의 엽록체 유전체는 interted repeat unit (IRa and Irb)에 의해 Large single copy region (LSC)과 small single copy region (SSC)으로 나뉘어진다. 화살표 부분을 끊어 선형 DNA를 만든다면 자신의 서열에 대한 자신의 서열을 dot plot 하면 어떤 모습일까?

Guess the sequence composition of sequences A based on the following dot plot!



Tandem repeat sequence!

Recognition of WGD (Whole Genome Duplications) using self-dot plot of a genome

Arabidopsis thaliana →

2003년 *Arabidopsis* 전체 유전체가 밝혀진 이후 이를 dot plot을 이용하여 분석하여 매우 흥미로운 사실을 밝혀냈다. 다섯 개의 염색체를 일렬로 배열한 후 자신에 대한 자신의 dot plot을 해 본 결과 서열의 중간중간에 + 기울기의, 그리고 – 기울기의 선들을 발견하였다. 이것은 전체 유전체의 많은 부분이 서로 비슷한 구간이 존재함을 의미한다. 예를 들어서 염색체 3번의 뒷부분은 염색체 2번의 뒷부분과 거의 같은 염기서열을 갖고 있다(α11). 그리고 염색체 2번의 중간부위는 염색체 1번의 첫 부분과 거꾸로된 매우 유사한 부위를 갖고 있다(α2). 이렇게 중복이 일어난 모든 부위(노란 박스)를 모두 합쳐보면 전체 유전체 부위의 약60~70%에 해당하는 부위가 중복이 되었음을 알 수 있다. 이것은 *Arabidopsis*의 진화의 역사에서 한 지점에서 whole genome duplication (WGD; 전체 유전체 중복현상)이 일어나고 이후 부분적으로 치환, 결실, 중복, 전이 등이 일어났음을 암시한다. 노란 박스로 표시된 중복부위(α1~α27)을 일렬로 배열한 후 다시 자신에 대한 자신의 dot plot을 해 보면 다시 이들 내에서의 중복 구간을 인식할 수 있다(β로 표시된 노란 box). 이들 β box의 염기서열들을 다시 일렬로 배열한 후 dot plot을 하면 또다시 중복 부위를 나타내는 box들을 인식할 수 있었다. 이 논문에서는 전체 유전체의 단순한 dot plot 에 의해 피지식물이 나타난 이후 Arabidopsis까지의 진화 역사 동안에 적어도 3회 이상의 WGD 현상이 일어났음을 보여주고 있다.

http://amborella.net/2010Bioinformatics/Week04-Bowers%20et%20al%202003%20Nature%20Genome%20dup.pdf

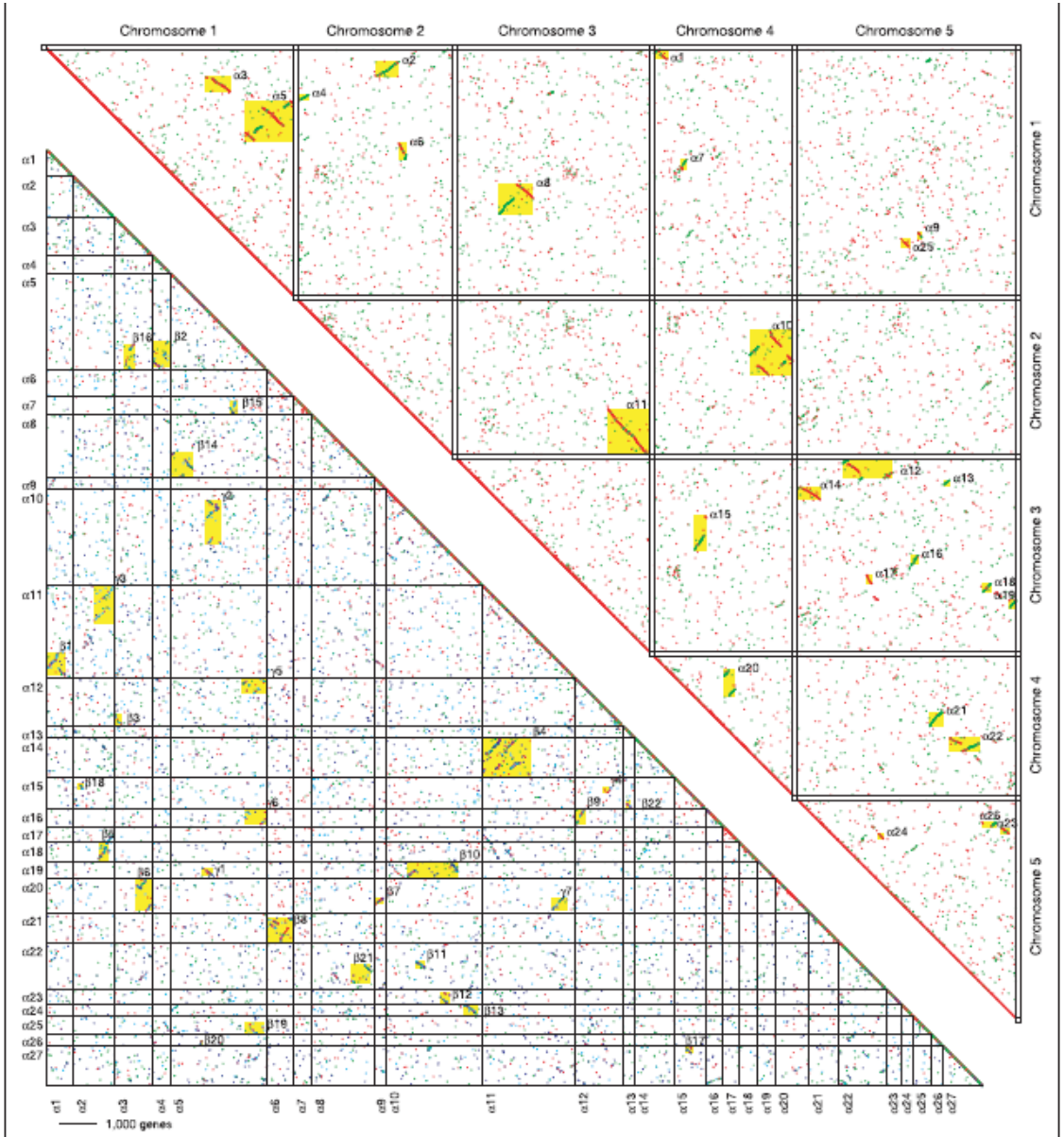
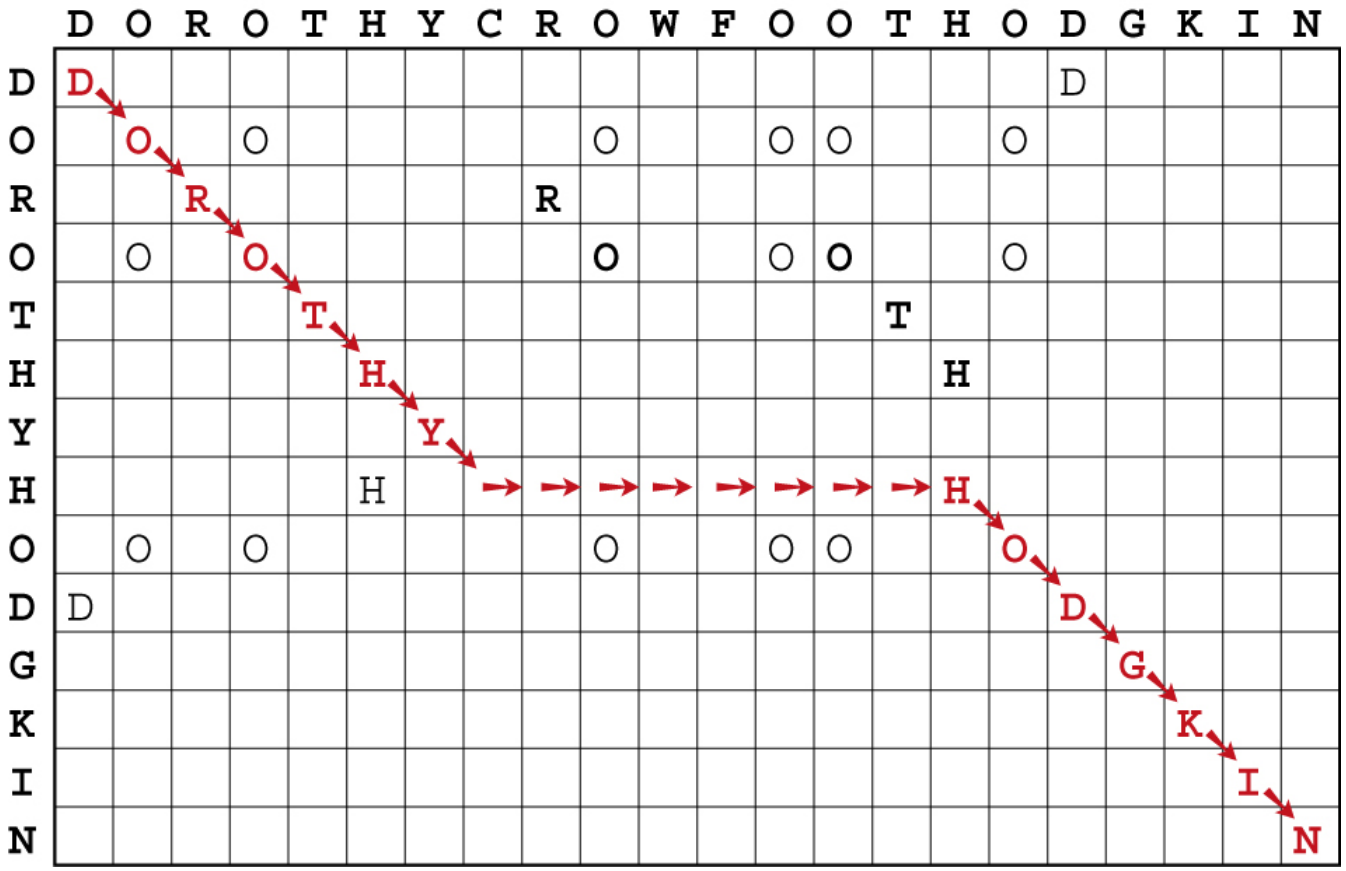


Figure 1 Arrangement of duplicated protein-encoding genes in *Arabidopsis thaliana*. Top right, α duplications. Both *x* and *y* axes represent 26,028 genes in their chromosomal order. The best-matching gene pairs are plotted, colour-coded to indicate same (red) or opposite (green) transcriptional orientations. For further analysis, 57 adjacent duplicated regions with opposite orientation and order explicable by localized inversions were combined into 26 'large' duplications (α01–α26) that each included ≥1% (260) of the genes. Eight shorter duplications were pooled (α27). Lower left, β and γ duplications. Both *x* and *y* axes represent 21,749 genes, in an inferred ancestral order that accounts for the composition of the 26 large α duplications (at left and bottom). Twenty-nine β or γ duplications (see text) are highlighted. Colours show how the four modern *Arabidopsis* chromosome segments contribute to β or γ duplications, distinguishing contributions to the segments at left and bottom respectively from the: (1) lower-numbered chromosomes (red); (2) higher- and lower-numbered chromosomes (light blue); (3) lower- and higher-numbered chromosomes (dark blue); (4) higher-numbered chromosomes (green). Higher-resolution versions of the figure and lists of gene orders are available (see Supplementary Information).

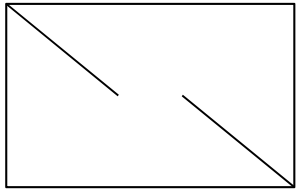
- 두개의 DNA 서열을 정렬한 3가지 결과
- A match
- A mismatch
- A gap + A insertion

ATACGGA	ATACGGA	ATACGGA
ATACGGA	GTACGGA	A_ACGGA



DOROTHYCROWFOOTHODKIN
DOROTHY-------HODKIN

GAP or indel



- Optimal alignment를 찾아내기는 매우 힘듬.

```
AGAT_G        AGATG
A_TACG        ___ATACG
```

- 무엇이 optimal alignment일까?
- 또 다른 alignment들이 있을까?

Simple alignment

- Match score and penalty

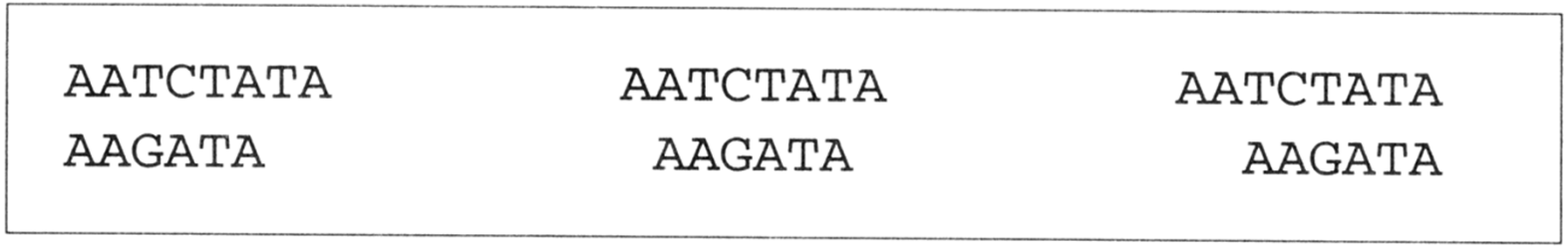


FIGURE 2.2 *Three possible simple alignments between two short sequences.*

$$\sum_{i=1}^{n} \begin{cases} \text{match score; if } seq1_i = seq2_i \\ \text{mismatch score; if } seq1_i \neq seq2_i \end{cases}$$

where n is the length of the longer sequence. For example, assuming a match score of 1 and a mismatch score of 0, the scores for the three alignments shown in Figure 2.2 would be 4, 1, and 3, from left to right.

Match score = 1
Mismatch score = 0 이라고 가정 할 때 위의 정렬서열들의 score 들은?

- 여러 종류의 정렬서열들이 있다면 어떤 것이 가장 좋은 것(optimal)인지 판단 할 기준이 필요함.

- Let's say if seq1 = '_' or seq2 = '_' : gap penalty
- If no gaps and seq1 = seq2 : match score
- If no gaps and seq1 not equal seq2 : mismatch score
- 만약

Match score = 1

Mismatch score = 0

Gap penalty = -1 이라고 가정하면…

Gaps in alignment

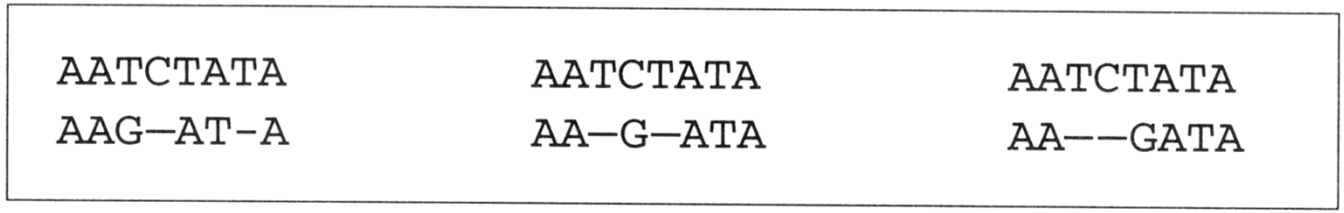


FIGURE 2.3 *Three possible gapped alignments between two short sequences.*

$$\sum_{i=1}^{n} \begin{cases} \text{gap penalty; if } seq1_i = \text{'–'} \text{ or } seq2_i = \text{'–'} \\ \text{match score; if no gaps and } seq1_i = seq2_i \\ \text{mismatch score; if no gaps and } seq1_i \neq seq2_i \end{cases}$$

Simple gap penalty:
Assuming match score = 1, mismatch score = 0, gap penalty = -1
라고 가정할 때 위의 정렬서열들의 score 들은?

- 위의 가정으로 두 sequence를 비교해 보자.
- AATCTATA 와 AAGATA

AATCTATA	AATCTATA	AATCTATA
AAG-AT-A	AA-G-ATA	AA--GATA
1+1+0	1+1+(-1)+0	1+1+(-1)+(-1)
+(-1)+0+0	+(-1)+1+	+0+1+1+1
+(-1)+1	+1+1	
=1	=3	=3

- Score를 보면 첫번째 alignment는 optimal 한 것이 아닌 것으로 보임.
- 두번째 세번째 alignment와 같이 score 가 같은 경우에는 어떤 것을 선택해야 하나????
- 우리는 하나의 nucleotide 의 insertion 이나 deletion 이 여러 개의 insertion이나 deletion이 일어날 확률보다 더 높다는 것을 알고 있음.
- Origination penalty – gap의 시작
- Length penalty – gap이 이어지는 것

Origination of gaps

- Insertion vs. deletion (indel) events
  = one step event in evolution
- Origination penalty (open gap penalty)
  → higher penalty value
- Length penalty (gap extension penalty)
  → smaller penalty value

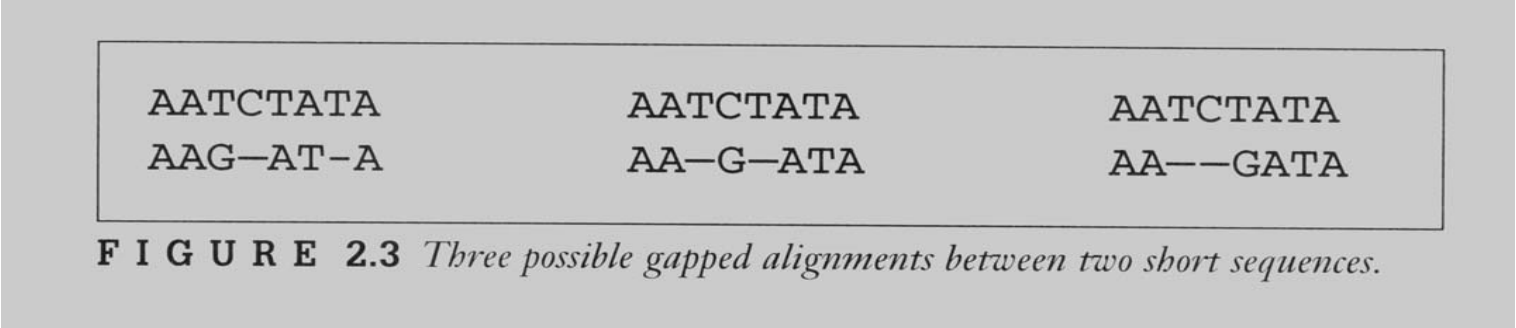


FIGURE 2.3 *Three possible gapped alignments between two short sequences.*

Gap origination penalty= -2, length penalty= -1,
Match score= 1, mismatch score= 0

- origination penalty를 만들기 보다는 length penalty를 더 많이 만드는 것이 더 좋다고 생각됨.
- 예를 들어 origination penalty = -2

  Length penalty = -1 라 가정하면

AATCTATA	AATCTATA	AATCTATA
AAG-AT-A	AA-G-ATA	AA--GATA
1+1+0+(-2)+0+0+(-2)+1 = -1	1+1+(-2)+0+(-2)+1+1+1 = +1	1+1+(-2)+(-1)+0+1+1+1 = +2

Scoring matrices – nucleotide (정렬점수표)

Identity Matrix

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

BLAST Matrix

	A	T	C	G
A	5	–4	–4	–4
T	–4	5	–4	–4
C	–4	–4	5	–4
G	–4	–4	–4	5

Transition Transversion Matrix

	A	T	C	G
A	1	–5	–5	–1
T	–5	1	–1	–5
C	–5	–1	1	–5
G	–1	–5	–5	1

transversion

transition

FIGURE 2.4 *Scoring matrices for aligning DNA sequences.*

- 진화에 있어서 보존적인 치환이 일어날 가능성이 더 많다.
- 일치와 불일치의 score를 다르게 해 줄 수 도 있고,
- 모든 불일치가 다 같은 penalty를 갖는 것이 아니라 정도를 달리 해 주는 경우도 있음 → weighted score

Scoring matrices – amino acid residues

- Amino acid scoring matrix를 만드는 두 가지 방법

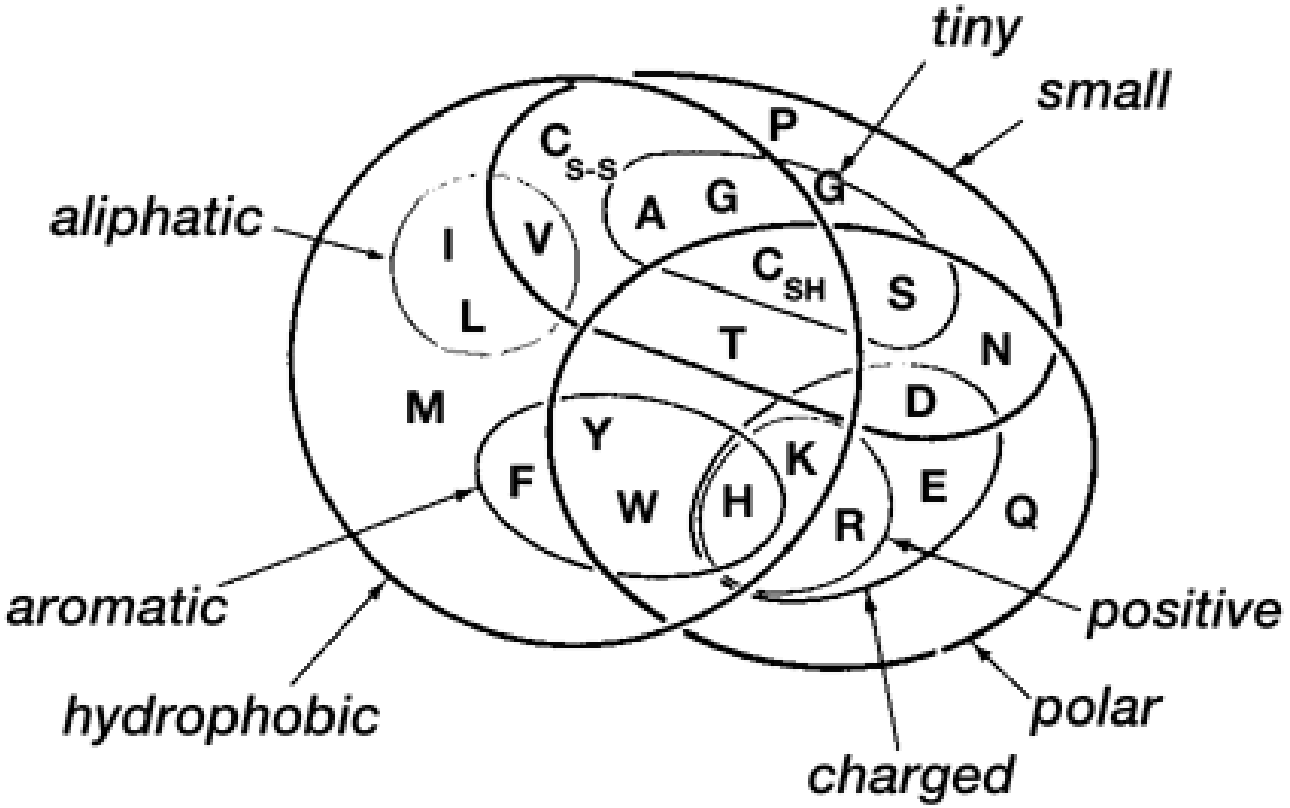
1) 아미노산들 사이의 화학적, 물리적 유사도에 기반하여 가중치를 줌.
   → 너무나 다양한 물리적, 화학적 유전적 factor들이 존재하기 때문에 객관적인 표현이 매우 힘듬

2) 실제 관찰에 의해 아미노산간의 치환빈도를 조사하여 적용

Scoring matrices – amino acid residues

- PAM (point accepted mutation) matrix – 매우 유사한 단백질서열 들을 정렬했을 때의 관찰된 치환을 계산한 값들(상대적 mutation 빈도)로 만듬
- PAM unit – 100개의 residue당 하나의 치환이 일어나기 위해 소요되는 시간
- 작은 수치의 PAM matrix는 보다 가까운 protein들의 비교에 이용됨.
- 일반적으로 PAM250 을 이용
- BLOSUM matrix – gap이 없는 유사한 protein sequence들에 대하여 통계학적인 clustering method를 적용 후, 군집간의 치환율을 계산하여 만듬
- 높은 수치의 BLOSUM matrix가 가까운 protein 서열들의 비교에 이용됨.
- 일반적으로 BLOSUM 62을 이용.
  → 약 62%의 similarity를 갖는 protein sequence들의 데이터로 만들어낸 matrix

Similarity: Physico-Chemical Properties of Amino Acids

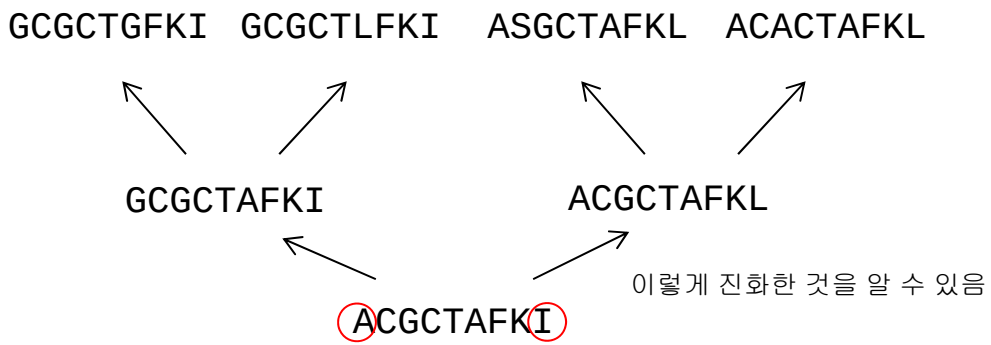


PAM-1 (Point Accepted Mutation) matrix 만들기

1. Multiple sequence alignment를 만듬.

```
ACGCTAFKI
GCGCTAFKI
ACGCTAFKL
GCGCTGFKI
GCGCTLFKI
ASGCTAFKL
ACACTAFKL
```

2. Matrix로 부터 phylogenetic tree를 만듬.
(계통수를 만드는 과정은 chpa 4, 5에…)



3. 각각의 amino acid type에 대하여 다른 amino acid로 바뀐 빈도(substitution Frequency; *Fij*)를 조사.
여기서는 변화의 방향성이 없는것으로 간주함. 즉 A→G와 G→A는 같은 빈도임.

$F_{G,A}$는 A→G G→A 모두 count.
So, $F_{G,A}=3$

4. 각각의 amino acid에 대하여 relative mutability (*mi*) 를 모두 구함.
A 에 관계된 mutation = 4
전체 mutation의 수 = 6
- 양방향 mutation이니 2를 곱합
- Frequency of occurrence (m_j): 전체 matrix에서 amino acid의 수는 63개이고 A는 10개 10/63=0.159.
- scaling factor =100 즉 100을 곱함
(PAM1는 100 residue당 1개의 변화를 나타냄)

m_A = [4 / (6X2)] X0.159 X 100
= 0.0209

5. Mutation probability (*Mij*) 계산
Mij = $M_{G,A}$ = (0.0209 X 3) / 4 = 0.0156

6. 실제 PAM matrix에 들어갈 수치(*Rij*)는
$R_{G,A}$ = log(*Mji* / *mj*)
= log(0.0156 / 0.159) ≒ -1.01

7. 같은 amino acid간의 변화는
M_{jj}=1-m_j 를 계산하여 이를 6번에 대입하여 *Rjj*를 구함
즉 *Rjj* = log(*Mjj* / *mj*)
= log((1-$M_{A,A}$) / m_A) ※ $M_{A,A}$ =(0.0209X2) / 3
= log((1-*0.0139*) / 0.0209) ≒ 1.67

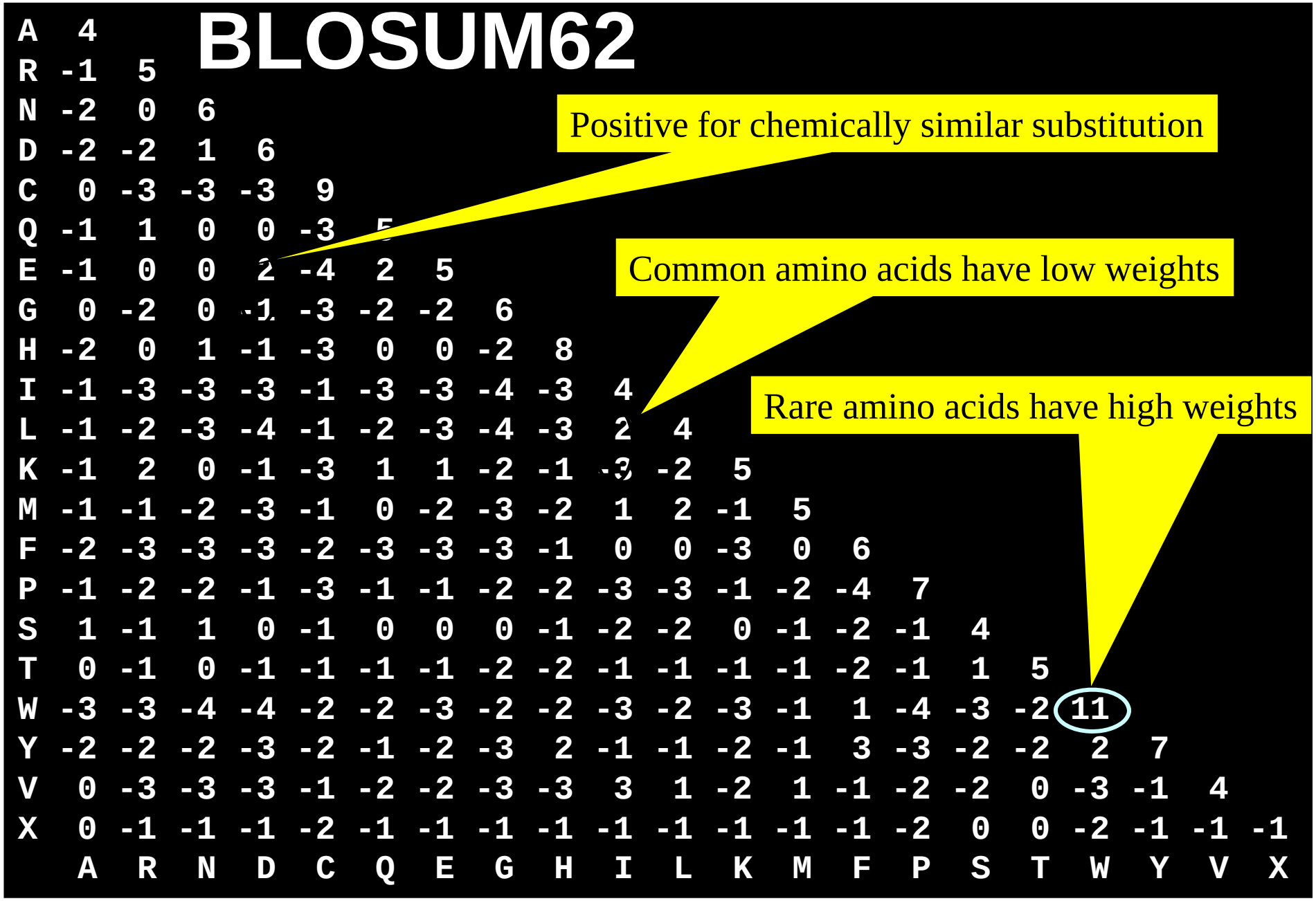
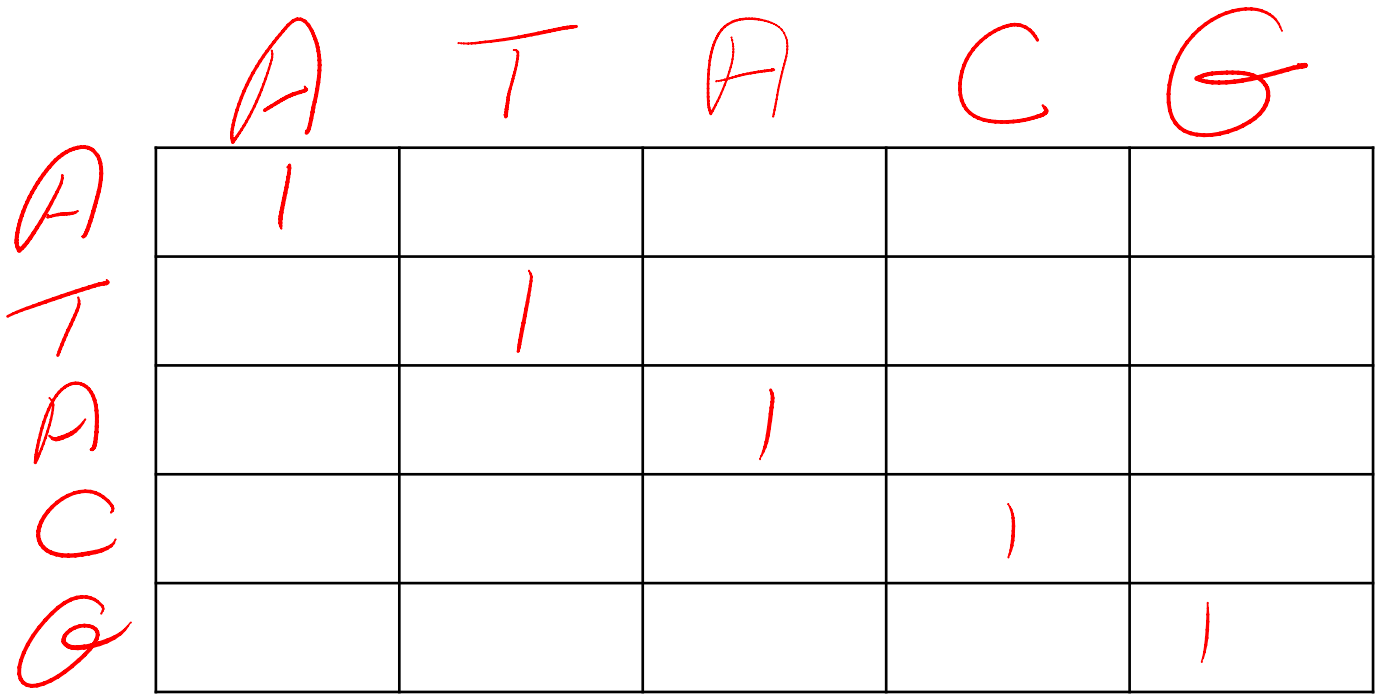

BLOSUM62
Positive for chemically similar substitution
Common amino acids have low weights
Rare amino acids have high weights
A 4
R -1 5
N -2 0 6
D -2 -2 1 6
C 0 -3 -3 -3 9
Q -1 1 0 0 -3 5
E -1 0 0 2 -4 2 5
G 0 -2 0 -1 -3 -2 -2 6
H -2 0 1 -1 -3 0 0 -2 8
I -1 -3 -3 -3 -1 -3 -3 -4 -3 4
L -1 -2 -3 -4 -1 -2 -3 -4 -3 2 4
K -1 2 0 -1 -3 1 1 -2 -1 -3 -2 5
M -1 -1 -2 -3 -1 0 -2 -3 -2 1 2 -1 5
F -2 -3 -3 -3 -2 -3 -3 -3 -1 0 0 -3 0 6
P -1 -2 -2 -1 -3 -1 -1 -2 -2 -3 -3 -1 -2 -4 7
S 1 -1 1 0 -1 0 0 0 -1 -2 -2 0 -1 -2 -1 4
T 0 -1 0 -1 -1 -1 -1 -2 -2 -1 -1 -1 -1 -2 -1 1 5
W -3 -3 -4 -4 -2 -2 -3 -2 -2 -3 -2 -3 -1 1 -4 -3 -2 11
Y -2 -2 -2 -3 -2 -1 -2 -3 2 -1 -1 -2 -1 3 -3 -2 -2 2 7
V 0 -3 -3 -3 -1 -2 -2 -3 -3 3 1 -2 1 -1 -2 -2 0 -3 -1 4
X 0 -1 -1 -1 -2 -1 -1 -1 -1 -1 -1 -1 -1 -1 -2 0 0 -2 -1 -1 -1
A R N D C Q E G H I L K M F P S T W Y V X

TABLE 11.1 ■ Selecting an Appropriate Scoring Matrix

Matrix	Best use	Similarity (%)
PAM40	Short alignments that are highly similar	70–90
PAM160	Detecting members of a protein family	50–60
PAM250	Longer alignments of more divergent sequences	~30
BLOSUM90	Short alignments that are highly similar	70–90
BLOSUM80	Detecting members of a protein family	50–60
BLOSUM62	Most effective in finding all potential similarities	30–40
BLOSUM30	Longer alignments of more divergent sequences	<30

The Similarity column gives the range of similarities that the matrix is able to best detect (c.f., Wheeler, 2003).

- A matrix is a grid used to analyze the optimal alignment of two DNA fragments



- A matrix is a grid used to analyze the optimal alignment of two DNA fragments

Algorithms for searching the best alignment

- 모든 경우를 다 따지는 것은 매우 힘든 일임.
- Dynamic programming – 최종결론을 얻기 위하여 하나의 문제를 작은 문제들로 나누어 부분 해결을 한 것을 이어 붙이는 방법
- 100nt와 95nt의 두 sequence를 정렬하려고 한다면

  → 모든 가능한 alignment들은

  approximately 55 million!
- 이러한 문제를 해결하기 위해 sequence를 작은 부분들로 나눔.

Needleman and Wunsch Algorithm

First Position	Score	Sequences Remaining to be Aligned
C C	+1	ACGA CGA
– C	–1	CACGA GA
C –	–1	ACGA CGA

FIGURE 2.5 *Three possibilities for aligning the first position in the sequences CACGA and CCGA. The match bonus is +1, the mismatch score is 0, and the gap penalty is –1.*

CACGA
CGA

-CACGA
CGA

CACGA
-CGA

1번 position의 정렬순서가 결정되었으므로 이를 제외한 나머지만 갖고 정렬가능.
이러한 방식으로 큰 문제를 작게 나누어 해결함.

- Matrix를 만들고 첫 번 째 줄과 열에 0, -1, -2, -3…의 숫자를 써 넣음 (gap penalty의 의미)
- (2,2) 위치에 대하여

1) 수직축에 위치한 서열에 공백감점(-1) 더한 값

2)수평축에 위치한 서열에 공백감점(-1) 더한 값

3)(2,2)에 위치한 염기들을 정렬하였을 때 일치가점(1) 또는 불일치감점(0)과 (1,1)의 값을 더한 값

1), 2), 3)을 비교하여 가장 큰 수를 선택

- 마찬가지로 계산하여 전체 matrix를 채움

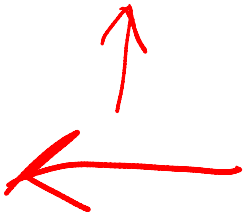
		A	C	T	C	G
	0	-1	-2	-3	-4	-5
A	-1					
C	-2					
A	-3					
G	-4					
T	-5					
A	-6					
G	-7					

		A	C	T	C	G
	0	-1	-2	-3	-4	-5
A	-1	1	0	-1	-2	-3
C	-2	0	2	1	0	-1
A	-3	-1	1	2	1	0
G	-4	-2	0	1	2	2
T	-5	-3	-1	1	1	2
A	-6	-4	-2	0	1	1
G	-7	-5	-3	-1	0	2

- Matrix를 완성 한 후 가장 아래 좌측으로부터
- 이 점수를 만든 것은 위인가 좌인가 상좌인가?
- 이 점수를 만든 위치로 화살표를 만듬.
- 수직화실표는 위의 서열에 gap이 있는 것임.
- 수평화실표는 좌의 서열에 gap이 있는 것임.
- Final alignment

AC--TCG

ACAGTAG




A C T C G
A C A G T A G
0 -1 -2 -3 -4 -5
-1 1 0 -1 -2 -3
-2 0 2 1 0 -1
-3 -1 1 2 1 0
-4 -2 0 1 2 2
-5 -3 -1 1 1 2
-6 -4 -2 0 1 1
-7 -5 -3 -1 0 2