Gaps

Kim ChangGi

2015. 3. 17

Introduction to Gaps

- **Definition**

  - A gap is any maximal, consecutive run of spaces in a single string of a given alignment

Introduction to Gaps

- **Example**

C	T	T	T	A	A	C	-	-	A	-	A	C
C	-	-	-	C	A	C	C	C	A	T	-	C

- **4 Gaps with 7 spaces**

- **Why not 3 Gaps?**

Introduction to Gaps

- **Example**

C	T	T	T	A	A	C	-	-	A	-	A	C
C	-	-	-	C	A	C	C	C	A	T	-	C

- **4 Gaps with 7 spaces**

- **Why not 3 Gaps?**
  - Contiguous deletion & insertion are counted as different gap.

Introduction to Gaps

- **Simplest object function (include gaps)**

  -each gaps contribute W_g, independent of how long the gap is

  (so that ‘s(*x*, _) = s(_, *x*) = 0’)

$$\sum_{i=1}^{l} s\left(S'_1(i),\, S'_2(i)\right) - kW_g$$

Introduction to Gaps

- **Example**

s	*a*	*b*	*c*	*d*	-
a	1	-3	-2	0	-1
b		3	-2	-1	-2
c			0	-4	-2
d				3	-1
-					0

$S`_1$	c	a	c	-	d	b	d
$S`_2$	c	-	-	b	d	b	-
score	0	-1	-2	-2	3	3	-1

Score = 0

Introduction to Gaps

- **Example**

s	*a*	*b*	*c*	*d*	-
a	1	-3	-2	0	0
b		3	-2	-1	0
c			0	-4	0
d				3	0
-					0

$S`_1$	c	a	c	-	d	b	d
$S`_2$	c	-	-	b	d	b	-
score	0	0	0	0	3	3	0

Score = 6 - $3W_g$

Why gaps?

- **In many biological application, Insertion or Deletion is important**
  - Mutation is important, and gaps are justification of mutation.
  - Because mutations make insertion or deletion.

- **Example for long insertion or deletion**
  - unequal crossing over in meiosis
  - DNA slippage during replication
  - insertion of transposable element into a DNA string
  - insertion of DNA by retroviruses
  - translocations of DNA between chromosomes

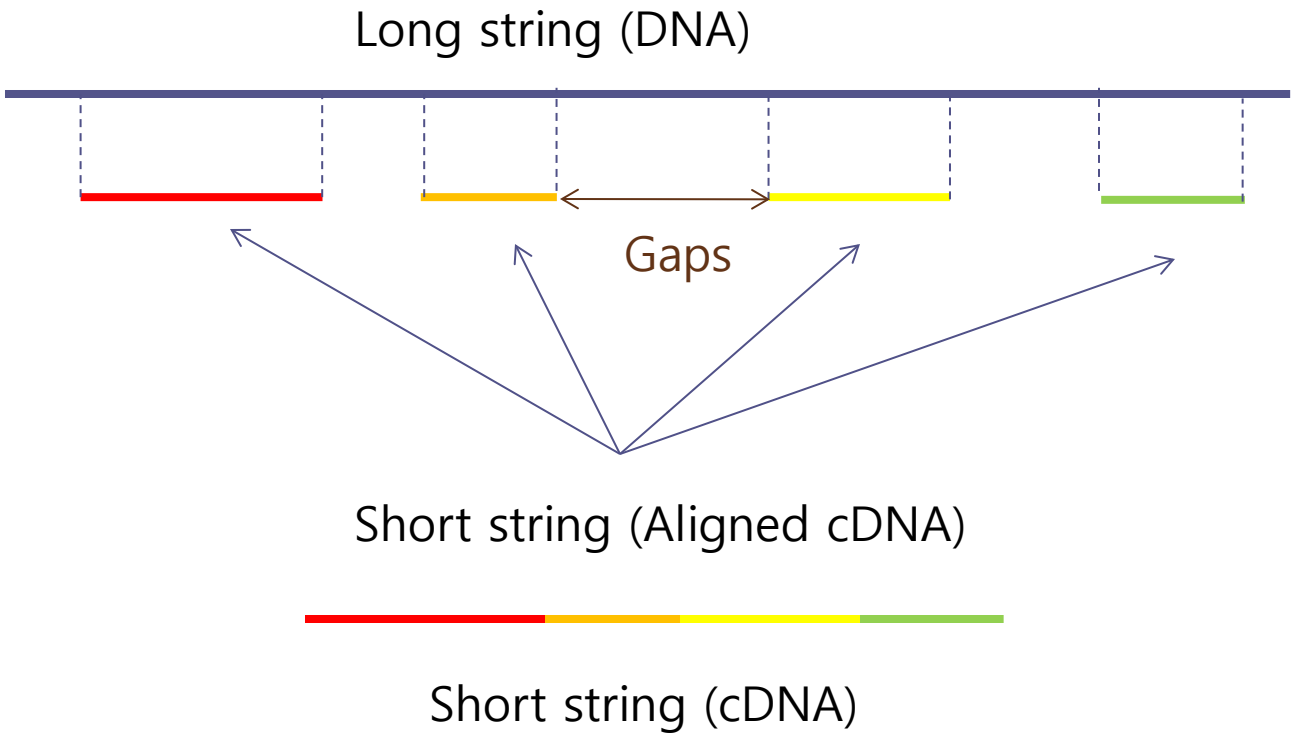
cDNA matching

- **What is cDNA**
  -in generate a protein, an RNA molecule is transcribed from the DNA of the gene
  : it calls mRNA
  - and hunt for the location of the gene, mRNA is used to create a DNA
  : it calls cDNA

- **Exon**
  - contribute to the code for the protein
  - Short
- **Intron**
  - intervening sequence (does not include code for protein)
  - Long

cDNA matching

- **Example**



cDNA matching

- **You don`t want to set a large penalty for spaces**
  -if not, it would align all string close together

- **Also, want a rather high penalty for mismatches**
  -when cDNA is correctly cut up, there are small percentage of mismatchs

- **Of course, positive match value**
  -need explanation?

cDNA matching

- **Longest common subsequence problem occur**
  - DNA has only four letter in roughly equal amount
  - intron is too longer than exon
  - some sequencing error
  - under this condition, all of the characters in cDNA exactly match with DNA stirng

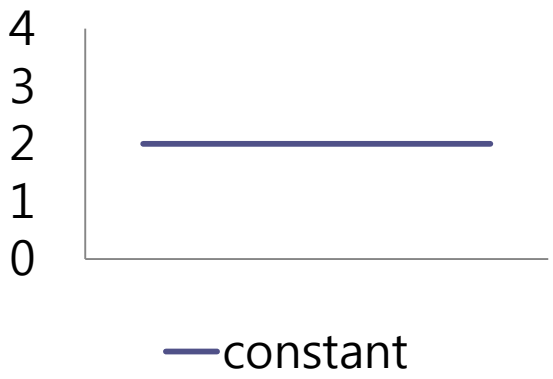
- **Under this condition cDNA matches all of the character in DNA**

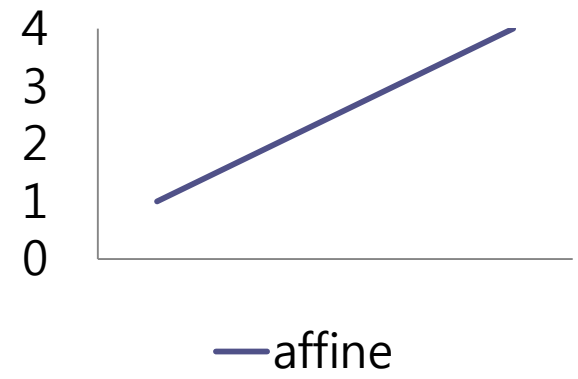
Choices for gap weights

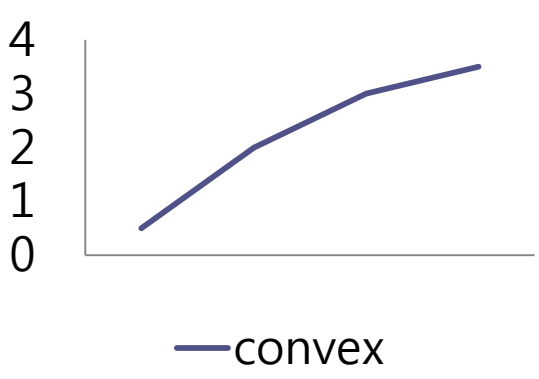
- **Examine in detail four general types of gap weight**
  - constant
  - affine
  - convex
  - arbitrary

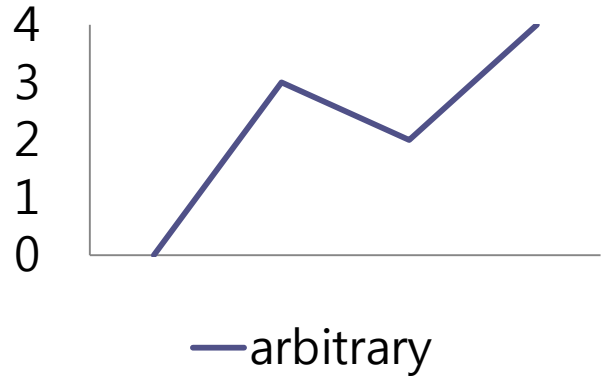
Choices for gap weights

- **Examine in detail four general types of gap weight**
  - constant
  - affine
  - convex
  - arbitrary









Choices for gap weights

- **Constant gap**
  - space is free
  - each gap is given a weight of W_g, independent of the number of spaces in the gap
  - W_m for match, W_{ms} for mismatch

Choices for gap weights

- **Constant gap**
  - Find an alignment A to maximize

$$[W_m(\#\text{matches}) - W_{ms}(\#\text{mismatches}) - W_g(\#\text{gaps})]$$

Choices for gap weights

- **Constant gap**
  - Find an alignment A to maximize

$$[W_m(\#\text{matches}) - W_{ms}(\#\text{mismatches}) - W_g(\#\text{gaps})]$$

- **If**
  - $W_m = 3$, $W_{ms} = 2$, $W_g = 1$

S`$_1$	b	c	a	-	-	f	d	g
S`$_2$	b	c	b	e	g	d	d	g

Choices for gap weights

- **Constant gap**
  - Find an alignment A to maximaize

$$[W_m(\#\text{matches}) - W_{ms}(\#\text{mismatches}) - W_g(\#\text{gaps})]$$

- **If**
  - $W_m = 3$, $W_{ms} = 2$, $W_g = 1$

S`$_1$	b	c	a	-	-	f	d	g
S`$_2$	b	c	b	e	g	d	d	g

$$W_m(\#\ matches) - W_{ms}(\#\ mismatches) - W_g(\#\ gaps)$$
$$= (3 * 4) - (2 * 2) - (1 * 1) = 12 - 4 - 1 = 7$$

Choices for gap weights

- **Constant gap**

  - Find an alignment A to maximize
    (adopt the alphabet-dependent weights for matches and mismatches)

$$\sum_{i=1}^{l} s(S'_1(i), S'_2(i)) - W_g(\#\text{gaps})$$

$$\cdot\ s(x, \_) = s(\_, x) = 0$$

· S'_1 and S'_2 represent the strings S_1 and S_2 after insertion spaces

Choices for gap weights

- **Examine in detail four general types of gap weight**
  - constant
  - affine
  - convex
  - arbitrary

Choices for gap weights

- **Affine gap weight**

  - add a weight W_s for each space in the gap
    - W_g (gap initiation weight) : cost of starting a gap
    - W_s (gap extension weight) : cost of extending the gap by one space

  - a single gap of length q is given by the affine function $W_g + qW_s$

  - generalization of the constant gap weight

  (constant gap weight model is simply the affine model with $W_s = 0$.)

Choices for gap weights

- **Affine gap weight**
  - Find an alignment A to maximize

$$[W_m(\#\text{matches}) - W_{ms}(\#\text{mismatches}) - W_g(\#\text{gaps}) - W_s(\#\text{spaces})]$$

- **If**
  - $W_m = 3$, $W_{ms} = 2$, $W_g = 1$, $W_s = 2$

S`$_1$	b	c	a	-	-	f	d	g
S`$_2$	b	c	b	e	g	d	d	g

W_m(# matches) - W_{ms}(# mismatches) - W_g(# gaps) - W_s (# spaces)

= (3 * 4) – (2 * 2) – (1 * 1) – (2 * 2) = 12 – 4 – 1 – 4 = 3

Choices for gap weights

- **Affine gap weight**
  - Find an alignment A to maximize
  (adopt the alphabet-dependent weights for matches and mismatches)

$$\sum_{i=1}^{l} s\left(S'_1(i), S'_2(i)\right) - W_g(\#\text{gaps}) - W_s(\#\text{spaces})$$

$$\cdot\ s(x, \_) = s(\_, x) = 0$$

$\cdot\ S'_1$ and S'_2 represent the strings S_1 and S_2 after insertion spaces

Choices for gap weights

- **Affine gap weight**
  - Find an alignment A to maximize
  (adopt the alphabet-dependent weights for matches and mismatches)

$$\sum_{i=1}^{l} s\left(S'_1(i), S'_2(i)\right) - W_g(\#\text{gaps}) - W_s(\#\text{spaces})$$

- If $W_g = 1, W_s = 2$

S'_1	c	a	c	-	-	b	d
S'_2	c	a	b	c	d	b	-
score	0						

s	**a**	**b**	**c**	**d**	**-**
a	1	-3	-2	0	0
b		3	-1	-4	0
c			0	-3	0
d				3	0
-					0

Choices for gap weights

- **Affine gap weight**
  - Find an alignment A to maximize
  (adopt the alphabet-dependent weights for matches and dismatches)

$$\sum_{i=1}^{l} s\left(S'_1(i), S'_2(i)\right) - W_g(\#\text{gaps}) - W_s(\#\text{spaces})$$

- If $W_g = 1, W_s = 2$

S'_1	c	a	c	-	-	b	d
S'_2	c	a	b	c	d	b	-
score	0	1	-1	0			

s	**a**	**b**	**c**	**d**	**-**
a	1	-3	-2	0	0
b		3	-1	-4	0
c			0	-3	0
d				3	0
-					0

Choices for gap weights

- **Affine gap weight**
  - Find an alignment A to maximize
  (adopt the alphabet-dependent weights for matches and dismatches)

$$\sum_{i=1}^{l} s\left(S'_1(i), S'_2(i)\right) - W_g(\#\text{gaps}) - W_s(\#\text{spaces})$$

- If $W_g = 1, W_s = 2$

S'_1	c	a	c	-	-	b	d
S'_2	c	a	b	c	d	b	-
score	0	1	-1	0	0	3	0

s	**a**	**b**	**c**	**d**	**-**
a	1	-3	-2	0	0
b		3	-1	-4	0
c			0	-3	0
d				3	0
-					0

$$(0 + 1 - 1 + 0 + 3 + 0) - (1 * 2) - (2 * 3) = -5$$

Choices for gap weights

- **Affine gap weight**

  - The affine gap weight model is the most commonly used gap model in the molecular biology literature

  - Example : program FASTA
    - FASTA is a DNA and protein sequence alignment software
    - default setting : $W_g = 10$, $W_s = 2$

  - more deeply section 11.8.6 (PPT 54 page)

Choices for gap weights

- **Examine in detail four general types of gap weight**
  - constant
  - affine
  - convex
  - arbitrary

Choices for gap weights

- **Convex gap weight**
  - used in some biological phenomena

  - each additional space in a gap contributes less to the gap weight than the preceding space

  -example : $W_g + \log q$

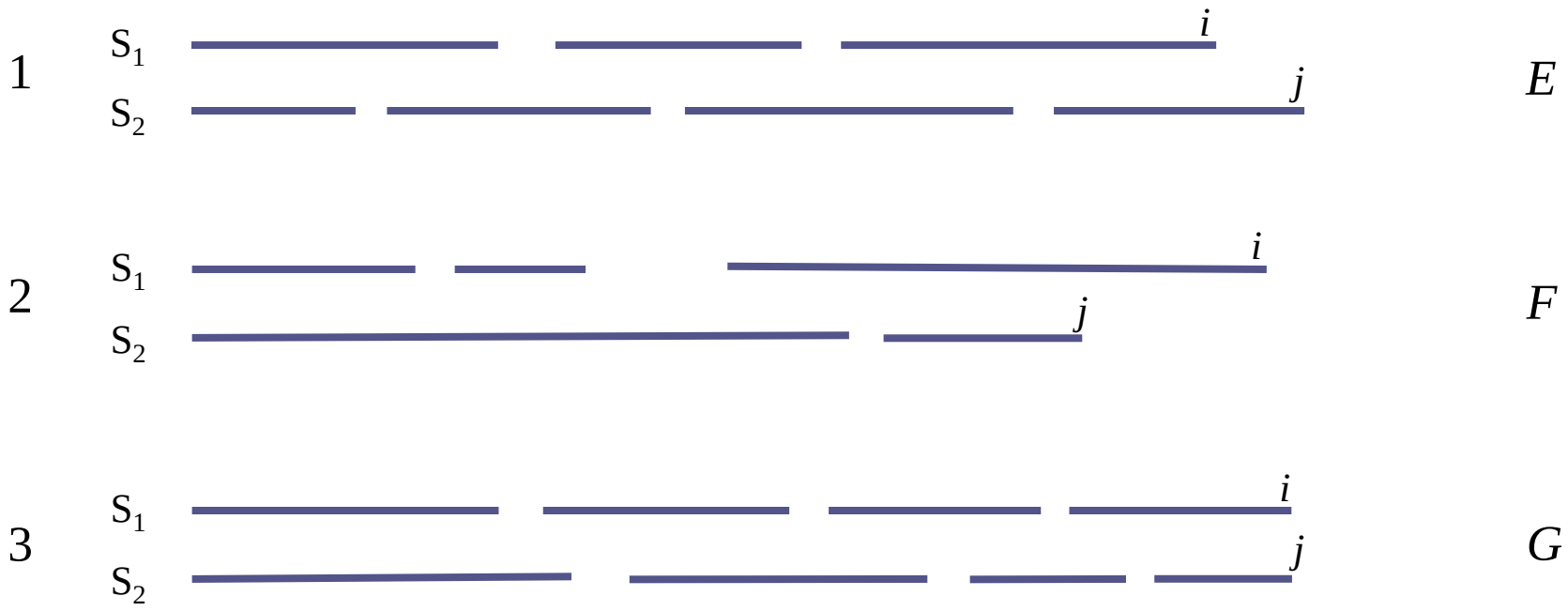
Arbitrary gap weights

- **Arbitrary gap weight are similar to Section 11.6.1**
  - but more detailed than Section 11.6.1

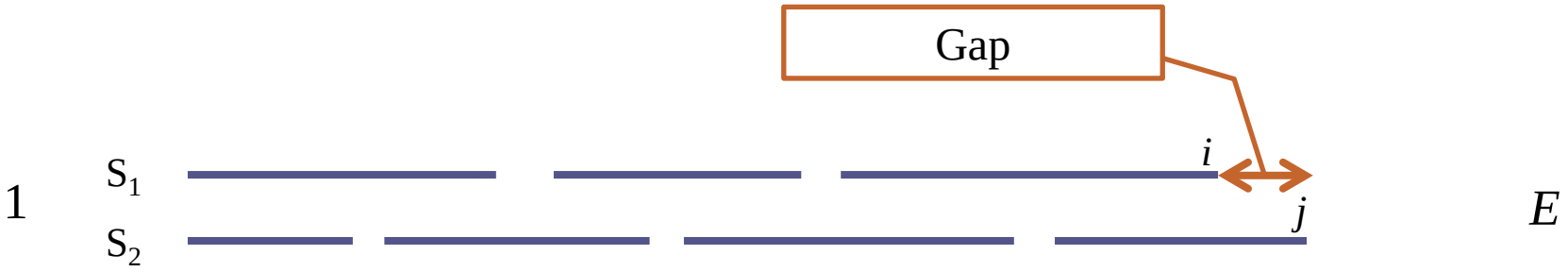
- **To alignment string S_1 and S_2**
  - prefixes $S_1[1 \ldots i]$ of S_1
  - prefixes $S_2[1 \ldots j]$ of S_2

Arbitrary gap weights

- **Any alignment of those two prefixes is one of the following three types.**

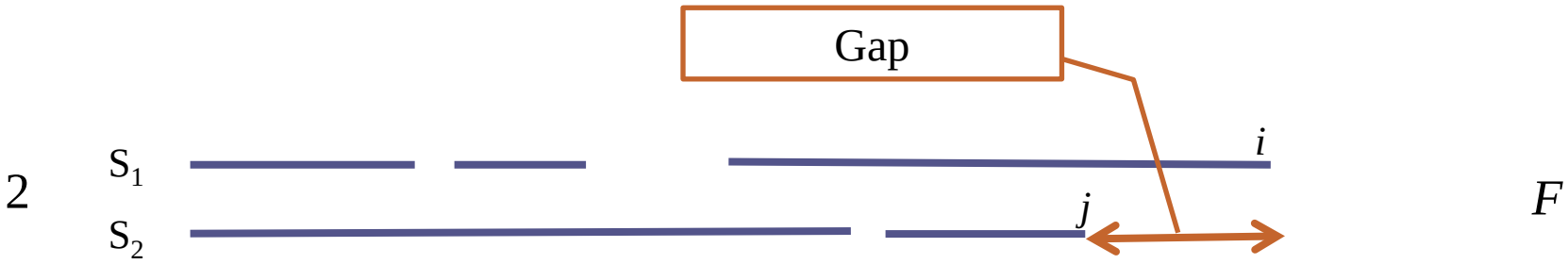


Arbitrary gap weights



- Character $S_1(i)$ is aligned to a character strictly, to the left of character $S_2(j)$
- The alignment ends with a gap in S_1
- **Define E(*i* , *j*) as the maximum value of any alignment of type 1;**

Arbitrary gap weights



- Character $S_1(i)$ is aligned to a character strictly, to the right of character $S_2(j)$
- The alignment ends with a gap in S_2
- **Define F(*i* , *j*) as the maximum value of any alignment of type 2;**

Arbitrary gap weights

3

- Characters $S_1(i)$ and $S_2(j)$ are aligned opposite each other.
- This includes both the case that $S_1(i) = S_2(j)$ and that $S_1(i) \neq S_2(j)$
- **Define G(*i* , *j*) as the maximum value of any alignment of type 3;**

Arbitrary gap weights

- **Definition**
  - Define E(*i* , *j*) as the maximum value of any alignment of type 1;
  - Define F(*i* , *j*) as the maximum value of any alignment of type 2;
  - Define G(*i* , *j*) as the maximum value of any alignment of type 3;
  - **Finally V(*i* , *j*) as the maximum value of the three terms E(*i* , *j*), F(*i* , *j*), G(*i* , *j*)**

Recurrences for the case of arbitrary gap weight

- **By dividing the types of alignments into three case**
  - We can write the following recurrences that establish V(i, j)

$$V(i,j) = \max\left[E(i,j), F(i,j), G(i,j)\right]$$

$$G(i,j) = V(i-1, j-1) + s(S_1(i), S_2(j))$$

$$E(i,j) = \max_{0 \le k \le j-1} \left[V(i,k) - w(j-k)\right]$$

$$F(i,j) = \max_{0 \le l \le i-1} \left[V(l,j) - w(i-l)\right]$$

Recurrences for the case of arbitrary gap weight

- **To complete the recurrences, we need to specify**
  - the base cases
  - and where the optimal alignment value is found

  - If all spaces are included in the objective function
    - base case is:

$$V(i, 0) = -w(i)$$
$$V(0, j) = -w(j)$$
$$E(i, 0) = -w(i)$$
$$F(0, j) = -w(j)$$

$G(0 , 0) = 0$ [but $G(i, j)$ is undefined if only j or i is 0]

Recurrences for the case of arbitrary gap weight

- **To complete the recurrences, we need to specify**
  - the base cases
  - and where the optimal alignment value is found

  - When end space and end gaps are free :

$$V(i\ ,\ 0) = 0$$

$$V(0\ ,\ j) = 0$$

Recurrences for the case of arbitrary gap weight

- **Example**

		c	a	b	c	b	d
	0	-8	-9	-1	7	9	2
c	-8	0	-8	-9	-1	4	9
a	-9	-8	1	-7	-2	0	1
c	-1	-9	-7	0	?		
b	7						
d	9						

Want to find V(4,3)

$$V(4,3) = \max [E(4,3), F(4,3), G(4,3)]$$

$$w(i) = 10 * \sin i$$

s	a	b	c	d	-
a	1	-3	-2	0	0
b		3	-1	-4	0
c			0	-3	0
d				3	0
-					0

Recurrences for the case of arbitrary gap weight

- **Example**

		c	a	b	c	b	d
	0	-8	-9	-1	7	9	2
c	-8	0	-8	-9	-1	4	9
a	-9	-8	1	-7	-2	0	1
c	-1	-9	-7	0	?		
b	7						
d	9						

$$G(4,3) = V(3,2) + s(S_1(3), S_2(2))$$

$$-7 + 0 = -7$$

s	**a**	**b**	**c**	**d**	**-**
a	1	-3	-2	0	0
b		3	-1	-4	0
c			0	-3	0
d				3	0
-					0

Recurrences for the case of arbitrary gap weight

- **Example**

		c	a	b	c	b	d
	0	-8	-9	-1	7	9	2
c	-8	0	-8	-9	-1	4	9
a	-9	-8	1	-7	-2	0	1
c	-1	-9	-7	0	?		
b	7						
d	9						

$$E(4,3) = \max_{0 \le k \le 2} [V(4,k) - w(3-k)]$$

Recurrences for the case of arbitrary gap weight

- **Example**

		c	a	b	c	b	d
	0	-8	-9	-1	7	9	2
c	-8	0	-8	-9	-1	4	9
a	-9	-8	1	-7	-2	0	1
c	-1	-9	-7	0	?		
b	7						
d	9						

$$E(4,3) = \max_{0 \le k \le 2} [V(4,k) - w(3-k)]$$

$$V(4,0) - w(3-0) = 7 + 1 = 8$$

c	a	b	c	-	-	-
-	-	-	-	c	a	c

Recurrences for the case of arbitrary gap weight

- **Example**

		c	a	b	c	b	d
	0	-8	-9	-1	7	9	2
c	-8	0	-8	-9	-1	4	9
a	-9	-8	1	-7	-2	0	1
c	-1	-9	-7	0	?		
b	7						
d	9						

$$E(4,3) = \max_{0 \le k \le 2} [V(4,k) - w(3-k)]$$

$$V(4,0) - w(3-0) = 7 - 1 = 6$$

$$V(4,1) - w(3-1) = -1 - 9 = -10$$

c	a	b	c	-	-
-	-	-	c	a	c

Recurrences for the case of arbitrary gap weight

- **Example**

		c	a	b	c	b	d
	0	-8	-9	-1	7	9	2
c	-8	0	-8	-9	-1	4	9
a	-9	-8	1	-7	-2	0	1
c	-1	-9	-7	0	?		
b	7						
d	9						

$$E(4,3) = \max_{0 \le k \le 2} [V(4,k) - w(3-k)]$$

$$V(4,0) - w(3-0) = 7 - 1 = 6$$

$$V(4,1) - w(3-1) = -1 - 9 = -10$$

$$V(4,2) - w(3-2) = -2 - 8 = -10$$

c	a	b	c	-
-	-	c	a	c

Recurrences for the case of arbitrary gap weight

- **Example**

		c	a	b	c	b	d
	0	-8	-9	-1	7	9	2
c	-8	0	-8	-9	-1	4	9
a	-9	-8	1	-7	-2	0	1
c	-1	-9	-7	0	?		
b	7						
d	9						

$$E(4,3) = \max_{0 \le k \le 2} [V(4,k) - w(3-k)]$$

$$V(4,0) - w(3-0) = 7 - 1 = 6$$

$$V(4,1) - w(3-1) = -1 - 9 = -10$$

$$V(4,2) - w(3-2) = -2 - 8 = -10$$

$$E(4,3) = \max[6, -10, -10] = 6$$

Recurrences for the case of arbitrary gap weight

- **Example**

		c	a	b	c	b	d
	0	-8	-9	-1	7	9	2
c	-8	0	-8	-9	-1	4	9
a	-9	-8	1	-7	-2	0	1
c	-1	-9	-7	0	?		
b	7						
d	9						

$$F(4,3) = \max_{0 \le l \le 3} [V(l,3) - w(4-l)]$$

Recurrences for the case of arbitrary gap weight

- **Example**

		c	a	b	c	b	d
	0	-8	-9	-1	7	9	2
c	-8	0	-8	-9	-1	4	9
a	-9	-8	1	-7	-2	0	1
c	-1	-9	-7	0	?		
b	7						
d	9						

$$F(4,3) = \max_{0 \le l \le 3}[V(l,3) - w(4-l)]$$

$$V(0,3) - w(4-0) = -1 + 7 = 6$$

$$V(1,3) - w(4-1) = -9 - 1 = -10$$

$$V(2,3) - w(4-2) = -7 - 9 = -16$$

$$V(3,3) - w(4-3) = 0 - 8 = -8$$

$$F(4,3) = \max[6, -10, -16, -8] = 6$$

Recurrences for the case of arbitrary gap weight

- **Example**

		c	a	b	c	b	d
	0	-8	-9	-1	7	9	2
c	-8	0	-8	-9	-1	4	9
a	-9	-8	1	-7	-2	0	1
c	-1	-9	-7	0	6		
b	7						
d	9						

$$V(4,3) = \max[-7, 6, 6] = 6$$

Time analysis

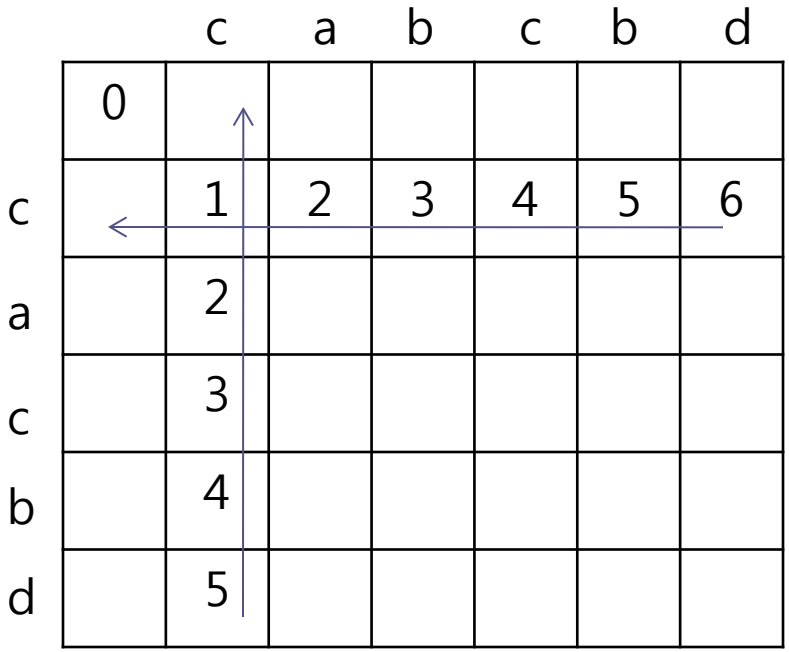
- **Theorem 11.8.1**
  - Assuming that $|S_1|=n$ and $|S_2|=m$
    - the recurrences can be evaluated in $O(nm^2+n^2m)$ time

Time analysis

- **Theorem 11.8.1**
  - Assuming that $|S_1|=n$ and $|S_2|=m$
    - the recurrences can be evaluated in $O(nm^2+n^2m)$ time
- **proof**
  - an $(n+1)$x$(m+1)$ size table are filled from left to right , up to down
  - To fill a cell (i , j)
  - To evaluate $E(i , j)$ examines j cells of row i,
    - $m(m+1)/2 = O(m^2)$ to evaluate E for that row n => $O(nm^2)$
  - To evaluate $F(i , j)$ examines i cells of column j,
    - $n(n+1)/2 = O(n^2)$ to evaluate F for that column m => $O(n^2m)$
  - To evaluate $G(i , j)$ examines one other cell => $O(nm)$
- **Since there are n rows and m columns give $O(nm^2+n^2m)$**

Time analysis

- **Proof**



$m = 6, \quad n = 5$

Affine (and constant) gap weight

- **Recall that the objective is to find as alignment to maximize**

  [W_m(# matches) - W_{ms}(# *mismatches*) - W_g(# *gaps*) - W_s (# *spaces*)]

- We will use the same variables in arbitrary gap weight
  $V(i , j)$, $E(i , j)$, $F(i , j)$, $G(i , j)$

- **In the affine gap weight model $w(q+1) - w(q) = W_s$**
  - for any gap length q greater than 0

Recurrences for affine gap weights

- **The base Case where end gaps are included :**

$$V(i\ ,\ 0) = E(i\ ,\ 0) = -\ W_g - iW_s$$
$$V(0\ ,\ j) = F(0\ ,\ j) = -\ W_g - jW_s$$

- **If end spaces are free and end gaps are free**

$$V(i\ ,\ 0) = V(0\ ,\ j) = 0$$

Recurrences for affine gap weights

- **Following recursive case:**
  - $V(i, j) = \max [E(i, j), F(i, j), G(i, j)]$
    - $G(i, j) = V(i\text{-}1, j\text{-}1) + W_m$ (if S1 = S2)
    - $G(i, j) = V(i\text{-}1, j\text{-}1) + W_{ms}$ (if S1 ≠ S2)
    - $E(i , j) = \max [E(i, j\text{-}1), V(i, j\text{-}1) - W_g] - W_s$
      - S_1 end with a gap
    - $F(i , j) = \max [F(i\text{ -}1, j), V(i\text{ -}1, j) - W_g] - W_s$
      - S_2 end with a gap

Recurrences for the case of arbitrary gap weight

- **Example**

		c	a	b	c	b	d
	0	-3	-4	-5	-6	-7	-8
c	-3	0	-3	-4	-5	-6	-7
a	-4	-3	1	-2	-3	-4	-5
c	-5	-4	-5	-6	?		
b	-6						
d	-7						

Want to find V(4,3)

$$V(4,3) = \max [E(4,3), F(4,3), G(4,3)]$$

$W_g = 2, W_s = 1$

s	a	b	c	d
a	1	-3	-2	0
b		3	-1	-4
c			0	-3
d				3
-				

Recurrences for the case of arbitrary gap weight

- **Example**

		c	a	b	c	b	d
	0	-3	-4	-5	-6	-7	-8
c	-3	0	-3	-4	-5	-6	-7
a	-4	-3	1	-2	-3	-4	-5
c	-5	-4	-5	-6	?		
b	-6						
d	-7						

$W_g = 2, W_s = 1$

$$G(4,3) = V(3,2) + s(S_1(3), S_2(2))$$
$$= -2 + 0 = -2$$

s	a	b	c	d
a	1	-3	-2	0
b		3	-1	-4
c			0	-3
d				3
-				

Recurrences for the case of arbitrary gap weight

- **Example**

		c	a	b	c	b	d
	0	-3	-4	-5	-6	-7	-8
c	-3	0	-3	-4	-5	-6	-7
a	-4	-3	1	-2	-3	-4	-5
c	-5	-4	-5	-6	?		
b	-6						
d	-7						

$W_g = 2, W_s = 1$

$$E(4,3) = \max[E(4,2), V(4,2) - 2] - 1$$
$$= \max[-8, -5] - 1$$
$$= -5 - 1 = -6$$

0	-3	-4	-5	-6	-7	-8
-3	-4	-5	-6	-7	-8	-9
-4	-5	-6	-7	-8	-9	-10
-5	-6	-7	-8	?		
-6						
-7						

Recurrences for the case of arbitrary gap weight

- **Example**

		c	a	b	c	b	d
	0	-3	-4	-5	-6	-7	-8
c	-3	0	-3	-4	-5	-6	-7
a	-4	-3	1	-2	-3	-4	-5
c	-5	-4	-5	-6	?		
b	-6						
d	-7						

$W_g = 2, W_s = 1$

$$F(4,3) = \max[F(3,3), V(3,3) - 2] - 1$$
$$= \max[-8, -8] - 1$$
$$= -8 - 1 = -9$$

0	-3	-4	-5	-6	-7	-8
-3	-4	-5	-6	-7	-8	-9
-4	-5	-6	-7	-8	-9	-10
-5	-6	-7	-8	?		
-6						
-7						

Recurrences for the case of arbitrary gap weight

- **Example**

		c	a	b	c	b	d
	0	-3	-4	-5	-6	-7	-8
c	-3	0	-3	-4	-5	-6	-7
a	-4	-3	1	-2	-3	-4	-5
c	-5	-4	-5	-6	-2		
b	-6						
d	-7						

$W_g = 2, W_s = 1$

$$V(4,3) = \max [E(4,3), F(4,3), G(4,3)]$$
$$= \max [-6, -9, -2]$$
$$= -2$$

Time analysis

- **Theorem 11.8.2**
  - The optimal alignment with affine gap weights can be computed in O(*nm*) time, the same time as for optimal alignment without a gap term

Time analysis

- **Theorem 11.8.2**
  - The optimal alignment with affine gap weights can be computed in O(*nm*) time, the same time as for optimal alignment without a gap term

- **proof**
  - an (*n*+1)x(*m*+1) size table are filled from left to right, up to down
  - each of cell calculate $E(i, j)$, $F(i, j)$, $G(i, j)$ for $V(i, j)$
  - E(i , j), F(i , j), G(i , j) are calculated in constant time

- **O(*nm*) time**