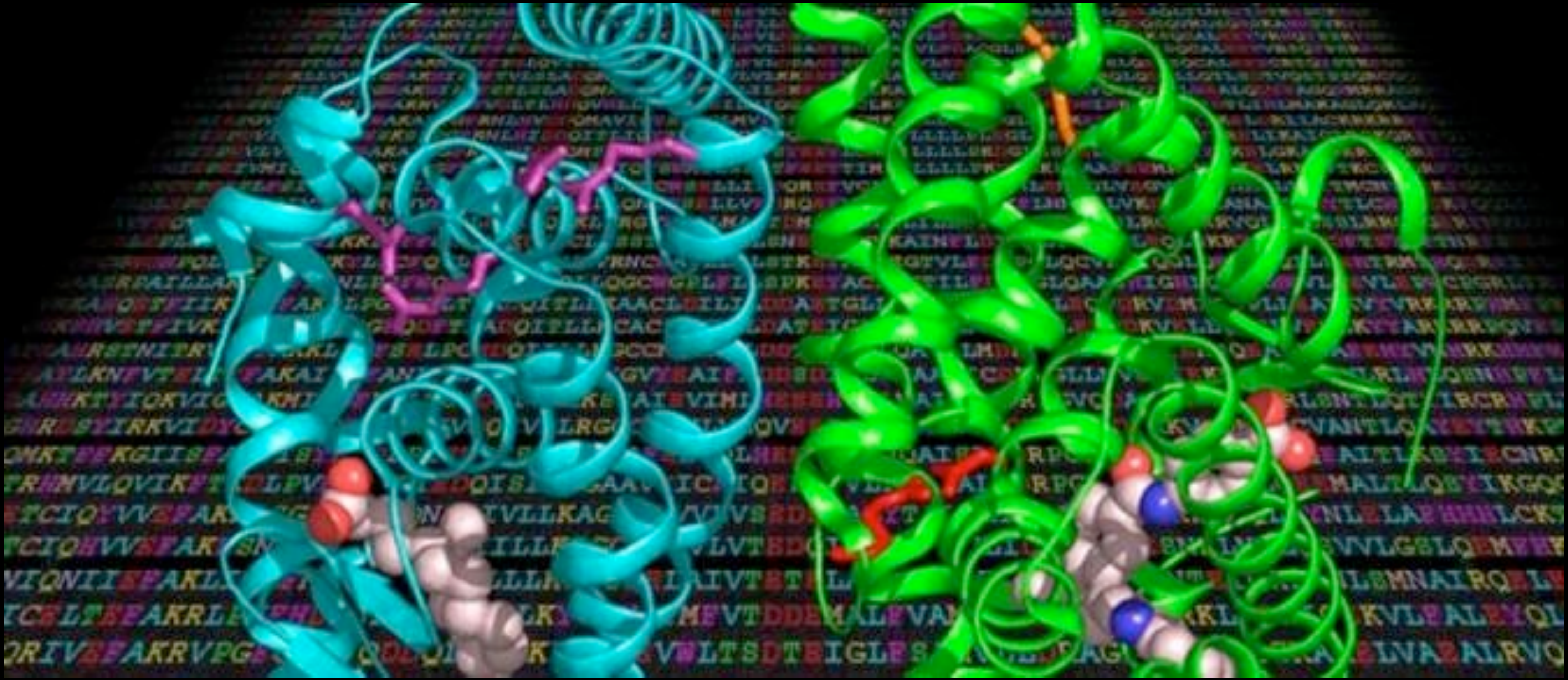
Dynamic Programming and Pairwise Sequence Alignment

Zahra Ebrahim zadeh
z.ebrahimzadeh@utoronto.ca



General Outline

- Importance of Sequence Alignment
- Pairwise Sequence Alignment
- Dynamic Programming in Pairwise Sequence Alignment
- Types of Pairwise Sequence Alignment

Importance of Sequence Alignment

- To identify **regions of similarity** :
  indicating functional and structural relationship

- To determine **homology**

What is pairwise sequence alignment?

```
IFCZ: S P Q L E E L I T K V S K A H Q E T F P - - - - - - - S L C Q L G K - -
3U9Q: S A D L R A L A K H L Y D S Y I K S F P L T K A K A R A I L T G K T T
```

The process of lining up two nucleotide or amino acid sequences to obtain highest score of similarity for the purpose of assessing the degree of similarity and possibility of homology

Alignment Operation

Transforming one string into the other by a series of edit operations on individual characters

Edit operations was first introduced in the edit distance concept by Levenshtein 1966.

- Insertion (I) of a character into the first string
- Deletion (D) of a character from the first string
- Substitution (S) of a character in the first string that mismatches the aligned character in the second string
- Match (M) of a character in the first strings with a character in the second string

Alignment Operation

Transforming one string into the other by a series of edit operations on individual characters

Edit operations was first introduced in the edit distance concept by Levenshtein 1966.

- Insertion (I) of a character into the first string
- Deletion (D) of a character from the first string
- Substitution (S) of a character in the first string that mismatches the aligned character in the second string
- Match (M) of a character in the first strings with a character in the second string

Example: V = THISLINE and W = ISALIGNED

V:	T	H	I	S	-	L	I	-	N	E	-
Edition on V:	D	D	M	M	I	M	M	I	M	M	I
W:	-	-	I	S	A	L	I	G	N	E	D

Difficulties in measuring sequence similarities

- Sequences usually differ in length

- Sequences may only have very small region of similarity

- Some substitution are more likely than others

Efficient way to find a best alignment

Consider aligning two sequences V = ($v_1v_2...v_n$) and W =($w_1w_2...w_m$).

Can we use Brute-Force method to create all the possible alignment, and then find the alignment with highest similarity score?

Efficient way to find a best alignment

Consider aligning two sequences V = ($v_1v_2...v_n$) and W =($w_1w_2...w_m$).

Can we use Brute-Force method to create all the possible alignment, and then find the alignment with highest similarity score?

This takes exponential time!

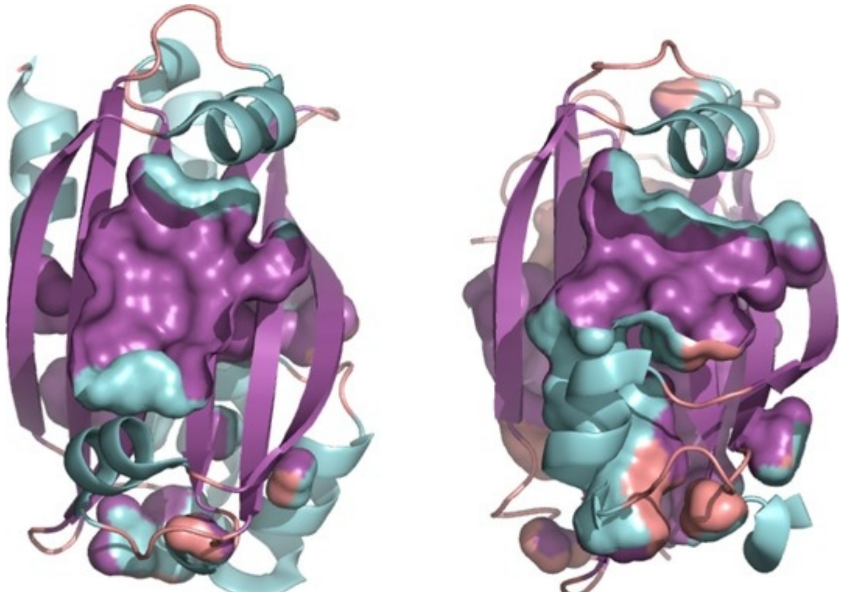
Efficient way to find a best alignment

Consider aligning two sequences V = $(v_1v_2...v_n)$ and W =$(w_1w_2...w_m)$.
Can we use Brute-Force method to create all the possible alignment, and then find the alignment with highest similarity score?

This takes exponential time!

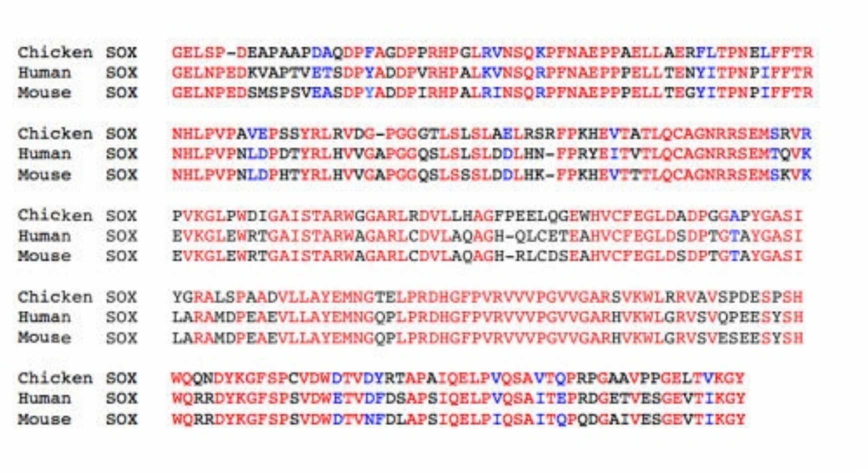
Dynamic Programming finds the optimal (best) alignment efficiently.

Problems Solvable by Dynamic Programming



Protein Structure Comparison

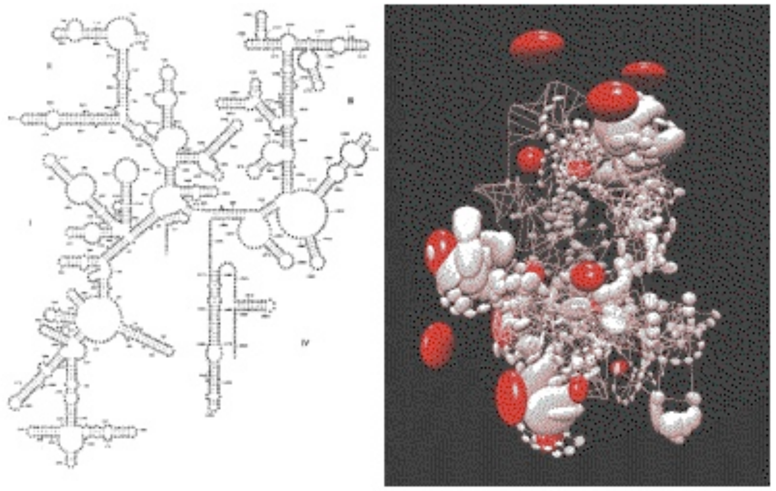
From: www.springerimages.com



```
Chicken SOX    GELSP-DEAPAAPDAQDPFAGDPPRHPGLRVNSQKPFNAEPPAELLAERFLTPNELFFTR
Human   SOX    GELNPEDKVAPTVETSDPYADDPVRHPALKVNSQRPFNAEPPPELLTENYITPNPIFFTR
Mouse   SOX    GELNPEDSMSPSVEASDPYADDPIRHPALRINSQRPFNAEPPPELLTEGYITPNPIFFTR

Chicken SOX    NHLPVPAVEPSSYRLRVDG-PGGGTLSLSLAELRSRFPKHEVTATLQCAGNRRSEMSRVR
Human   SOX    NHLPVPNLDPDTYRLHVVGAPGGQSLSLSLDDLHN-FPRYEITVTLQCAGNRRSEMTQVK
Mouse   SOX    NHLPVPNLDPHTYRLHVVGAPGGQSLSSSLDDLHK-FPKHEVTTTLQCAGNRRSEMSKVK

Chicken SOX    PVKGLPWDIGAISTARWGGARLRDVLLHAGFPEELQGEWHVCFEGLDADPGGAPYGASI
Human   SOX    EVKGLEWRTGAISTARWAGARLCDVLAQAGH-QLCETEAHVCFEGLDSDPTGTAYGASI
Mouse   SOX    EVKGLEWRTGAISTARWAGARLCDVLAQAGH-RLCDSEAHVCFEGLDSDPTGTAYGASI

Chicken SOX    YGRALSPAADVLLAYEMNGTELPRDHGFPVRVVVPGVVGARSVKWLRRVAVSPDESPSH
Human   SOX    LARAMDPEAEVLLAYEMNGQPLPRDHGFPVRVVVPGVVGARHVKWLGRVSVQPEESYSH
Mouse   SOX    LARAMDPEAEVLLAYEMNGQPLPRDHGFPVRVVVPGVVGARHVKWLGRVSVESEESYSH

Chicken SOX    WQQNDYKGFSPCVDWDTVDYRTAPAIQELPVQSAVTQPRPGAAVPPGELTVKGY
Human   SOX    WQRRDYKGFSPSVDWETVDFDSAPSIQELPVQSAITEPRDGETVESGEVTIKGY
Mouse   SOX    WQRRDYKGFSPSVDWDTVNFDLAPSIQELPIQSAITQPQDGAIVESGEVTIKGY
```

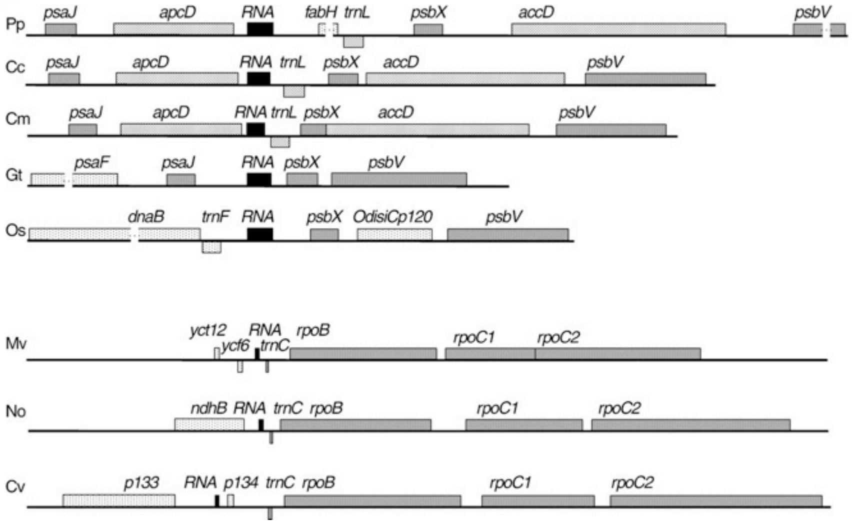
Sequence comparison

From: cbc.arizona.edu



RNA secondary structure prediction

From: palaeos.info



Gene Recognition

From: pcp.oxfordjournals.org

Dynamic Programming (DP)

A method for efficiently solving optimization problems which have overlapping subproblems

Property of DP problems

- Have overlapping subproblems
- Have optimal solutions to subproblems
- Can be represented in recurrence relation
- Are context-independent

e.g. In sequence alignment, quantifying similarity is only based on pairs of residues. Similarity is independent of environment of residues we align.

Classes of Pairwise Alignment:

Classes of Pairwise Alignment:

I. Global Alignment

Classes of Pairwise Alignment:

I. Global Alignment

Best match in the entire sequences

```
A  T  A  C  A  G  C  G  G  T  C  T
A  -  -  C  A  G  -  G  G  T  -  T
```

Classes of Pairwise Alignment:

I. Global Alignment

Best match in the entire sequences

```
A T A C A G C G G T C T
A - - C A G - G G T - T
```

II. Local Alignment

Classes of Pairwise Alignment:

I. Global Alignment

Best match in the entire sequences

```
A T A C A G C G G T C T
A - - C A G - G G T - T
```

II. Local Alignment

Best subsequence match

```
A T A C A G C G G T - C T
- - A C A G - G G T T - -
```

Classes of Pairwise Alignment:

I. Global Alignment

Best match in the entire sequences

```
A T A C A G C G G T C T
A - - C A G - G G T - T
```

II. Local Alignment

Best subsequence match

```
A T A C A G C G G T - C T
- - A C A G - G G T T - -
```

III. Semi-Global Alignment

"Glocal" Alignment

Classes of Pairwise Alignment:

I. Global Alignment

Best match in the entire sequences

```
A T A C A G C G G T C T
A - - C A G - G G T - T
```

II. Local Alignment

Best subsequence match

```
A T A C A G C G G T - C T
- - A C A G - G G T T - -
```

III. Semi-Global Alignment

"Glocal" Alignment

Best match without penalizing gaps on the ends of the alignment

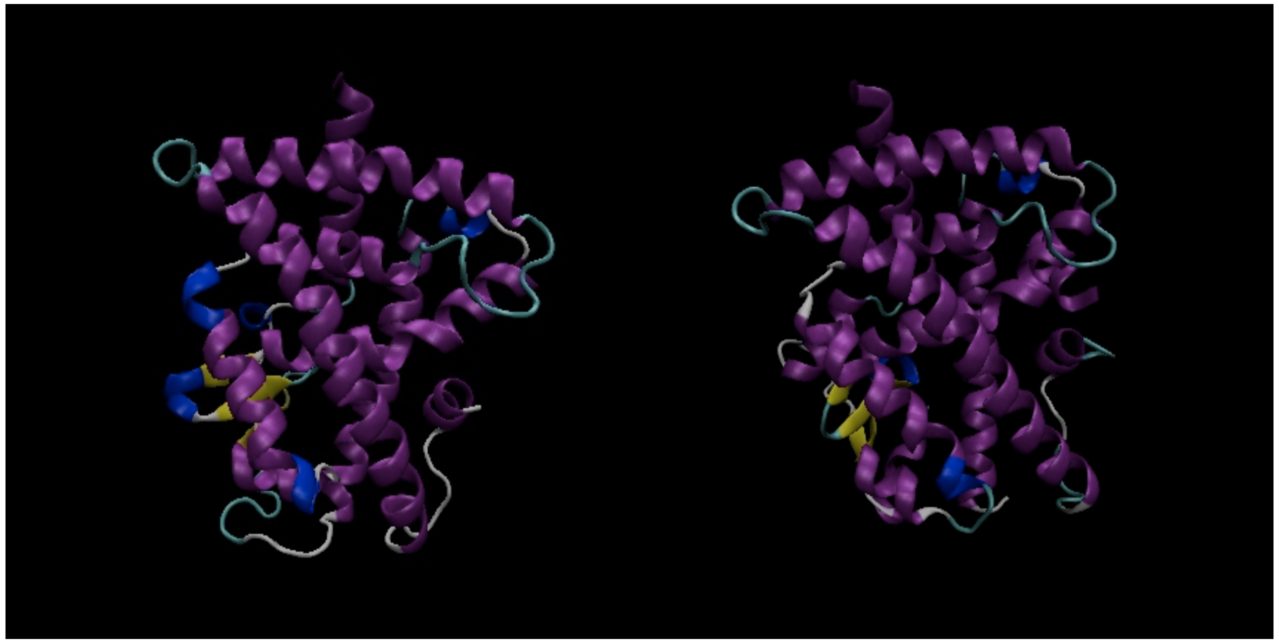
```
T C T G T - A C C G T G - -
- - - G T T A C C A T G C C
```

Global Alignment

spans all the residues in the sequences and finds best match in the entire two sequences

- Assumes the sequences are similar over the length of one another
- The alignment attempts to match them to each other from end to end

```
1FCZ: S P Q L E E L I T K V S K A H Q E T F P - - - - - - - S L C Q L G K - -
3U9Q: S A D L R A L A K H L Y D S Y I K S F P L T K A K A R A I L T G K T T
```



1FCZ 3U9Q

Optimal global alignments are produced using Needleman-Wunsch Algorithm

Needleman-Wunsch Algorithm

A dynamic programming algorithm for optimal global alignment

Given:

Two sequences V = ($v_1 v_2 \dots v_n$) and W =($w_1 w_2 \dots w_m$).

(|V| = n and |W|= m)

Goal:

Find the best scoring alignment in which all residues of both sequences are included. The score is usually a measure of similarity.

Requirement:

- A matrix NW of optimal scores of subsequence alignments.
  NW has size (n+1)x(m+1).
- Scoring matrix
- Defined gap penalty

Scoring matrix

represents a specific model of similarity to be applied in aligning two residues

- Matrix of numbers that quantify the similarity between residues

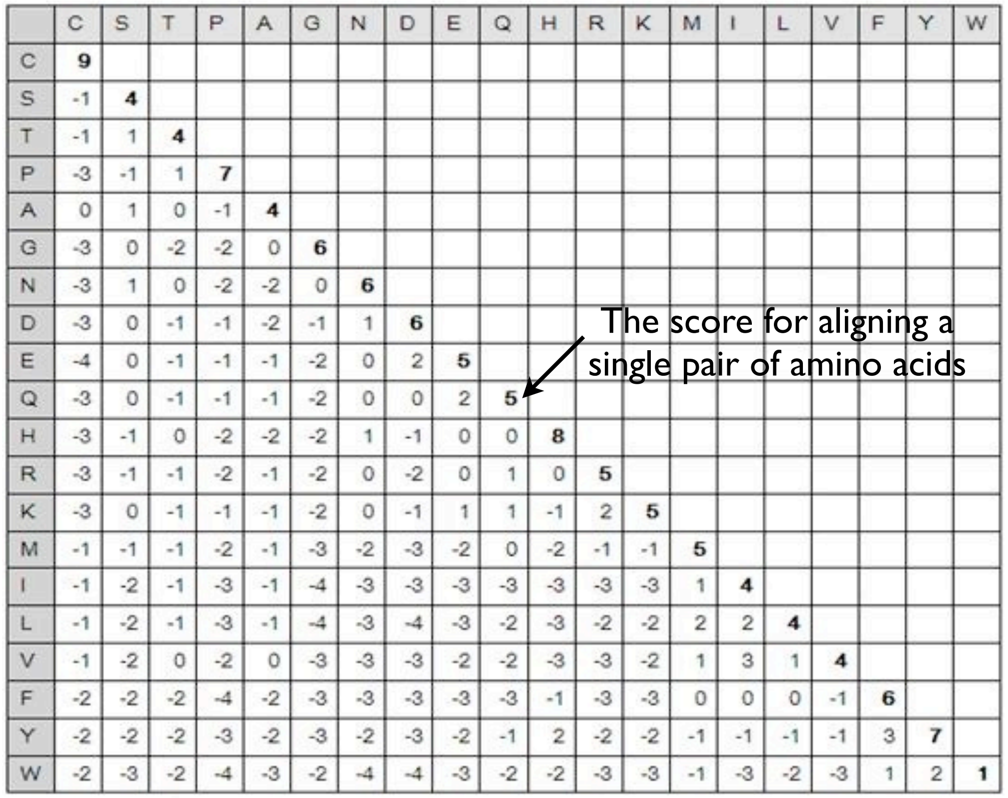
- To produce good alignment, the choice of a right scoring matrix is important

- Common scoring matrices:
    - Identity Matrix
    - Genetic Code Matrix
    - PAM Matrices
    - BLOSUM Matrices

- Protein sequences are frequently aligned using PAM or BLOSUM matrices that reflect the frequency with which a amino acid replaces another amino acid in evolutionarily related sequences.

    - Some amino acid substitutions are commonly found throughout the process of molecular evolution while others are rare.
    e.g. the probability that Ser mutates into Phe is ~ three times greater than the probability that Trp mutates into Phe

BLOSUM62



	C	S	T	P	A	G	N	D	E	Q	H	R	K	M	I	L	V	F	Y	W
C	**9**																			
S	-1	**4**																		
T	-1	1	**4**																	
P	-3	-1	1	**7**																
A	0	1	0	-1	**4**															
G	-3	0	-2	-2	0	**6**														
N	-3	1	0	-2	-2	0	**6**													
D	-3	0	-1	-1	-2	-1	1	**6**												
E	-4	0	-1	-1	-1	-2	0	2	**5**											
Q	-3	0	-1	-1	-1	-2	0	0	2	**5**										
H	-3	-1	0	-2	-2	-2	1	-1	0	0	**8**									
R	-3	-1	-1	-2	-1	-2	0	-2	0	1	0	**5**								
K	-3	0	-1	-1	-1	-2	0	-1	1	1	-1	2	**5**							
M	-1	-1	-1	-2	-1	-3	-2	-3	-2	0	-2	-1	-1	**5**						
I	-1	-2	-1	-3	-1	-4	-3	-3	-3	-3	-3	-3	-3	1	**4**					
L	-1	-2	-1	-3	-1	-4	-3	-4	-3	-2	-3	-2	-2	2	2	**4**				
V	-1	-2	0	-2	0	-3	-3	-3	-2	-2	-3	-3	-2	1	3	1	**4**			
F	-2	-2	-2	-4	-2	-3	-3	-3	-3	-3	-1	-3	-3	0	0	0	-1	**6**		
Y	-2	-2	-2	-3	-2	-3	-2	-3	-2	-1	2	-2	-2	-1	-1	-1	-1	3	**7**	
W	-2	-3	-2	-4	-3	-2	-4	-4	-3	-2	-2	-3	-3	-1	-3	-2	-3	1	2	**1**

The score for aligning a single pair of amino acids

Gap Penalty

a score for gap between the residues of sequences in sequence alignment

Gaps inserted in a sequence to maximize similarity with another, require a scoring penalty.

Gap opening penalty: penalty for starting a new gap in a sequence.

Gap extension penalty: penalty for adding gaps to an existing gap.

Common Gap Models:

- Constant gap: $g = -$ (gap opening penalty)
- Linear gap: $g(n_{gap}) = - n_{gap}$. (gap extension penalty)
- Affine gap: $w(n_{gap}) = -$ (gap opening penalty) - [n_{gap} . (gap extension penalty)] $= g + g(n_{gap})$

Affine gap model is used extensively in biology domain.

Needleman-Wunsch Algorithm (Cont.)

A dynamic programming algorithm for optimal global alignment

Given:

Two sequences V = ($v_1 v_2 \ldots v_n$) and W =($w_1 w_2 \ldots w_m$).

($|V| = n$ and $|W| = m$)

Goal:

Find the best scoring alignment in which all residues of both sequences are included. The score is usually a measure of similarity.

Requirement:

- A matrix NW of optimal scores of subsequence alignments. NW has size $(n+1)$x$(m+1)$.
- Score matrix
- Defined gap penalty

Needleman-Wunsch Algorithm (Cont.)

Calculation

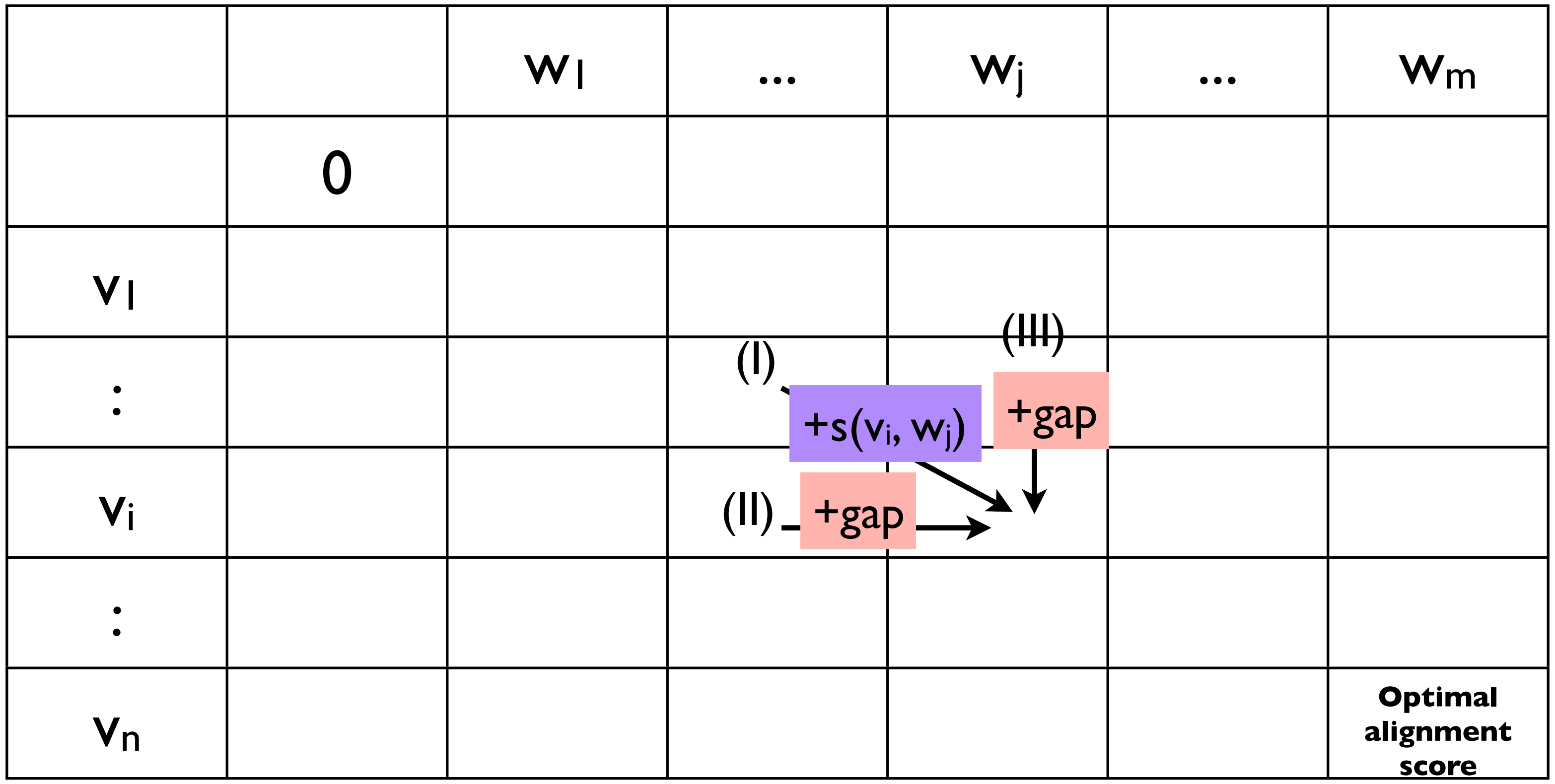
Let NW(i,j) be the optimal alignment score of aligning V[1…i] and W[1…j]

		w_1	…	w_j	…	w_m
	0					
v_1						
:			(I) +s(v_i, w_j)	(III) +gap		
v_i			(II) +gap			
:						
v_n						**Optimal alignment score**

Needleman-Wunsch Algorithm (Cont.)

Calculation

Let NW(i,j) be the optimal alignment score of aligning V[1…i] and W[1…j]



		w_1	…	w_j	…	w_m
	0					
v_1						
:						
v_i						
:						
v_n						**Optimal alignment score**

For linear gap penalty model

Base case:

$$\begin{cases} NW(0,0)=0 \\ NW(0,j)=NW(0,j-1)+g \\ NW(i,0)=NW(i-1,0)+g \end{cases}$$

Recurrence:

$$NW(i,j)=\max\begin{cases} NW(i-1,j-1)+s(v_i,w_j) \\ NW(i-1,j)+g \\ NW(i,j-1)+g \end{cases}$$

operations:

match/mismatch

delete

insert

Dynamic Programming Approach

Summary

Construct an optimal alignment between two subsequences ($v_1v_2...v_i$) and ($w_1w_2...w_j$), (Where $0 \leq i \leq n$ and $0 \leq j \leq m$), by considering the three cases:

(I) The optimal alignment of $v_{1,...,}v_{i-1}$ with $w_{1,...}w_{j-1}$, extended by the match between v_i and $w_{j.}$.

(II)The optimal alignment of $v_{1,...,}v_{i-1}$ with $w_{1,...}w_j$, extended by matching a gap character with v_i.

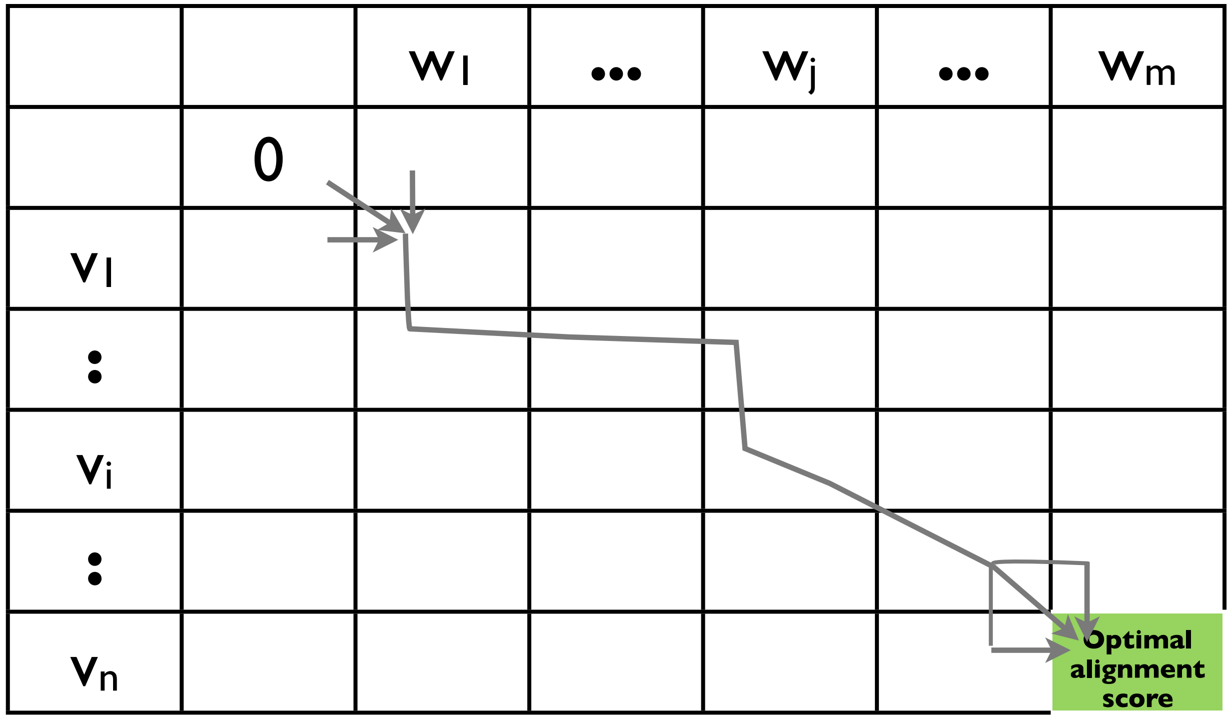
(III)The optimal alignment of $v_{1,...,}v_i$ with $w_{1,...}w_{j-1}$, extended by matching w_j with a gap character "-".

Store these optimal scores of subsequence alignments in a matrix of size (n+1) x (m+1).

Needleman-Wunsch Algorithm (Cont.)

Traceback

To recover the optimal alignment, arrows indicating forward calculation paths, are placed in each entry.

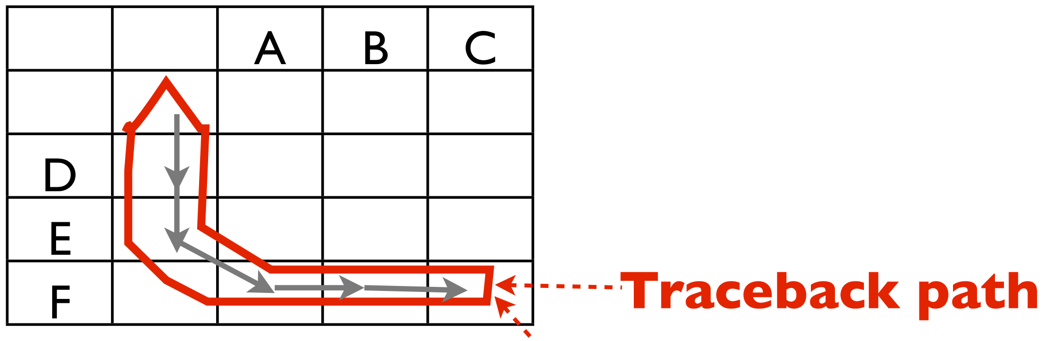


↘ : $NW(i,j) = NW(i-1,j-1) + s(v_i,w_j)$

↓ : $NW(i,j) = NW(i-1,j) + gap$

→ : $NW(i,j) = NW(i,j-1) + gap$

Determine alignment from the end of the sequences



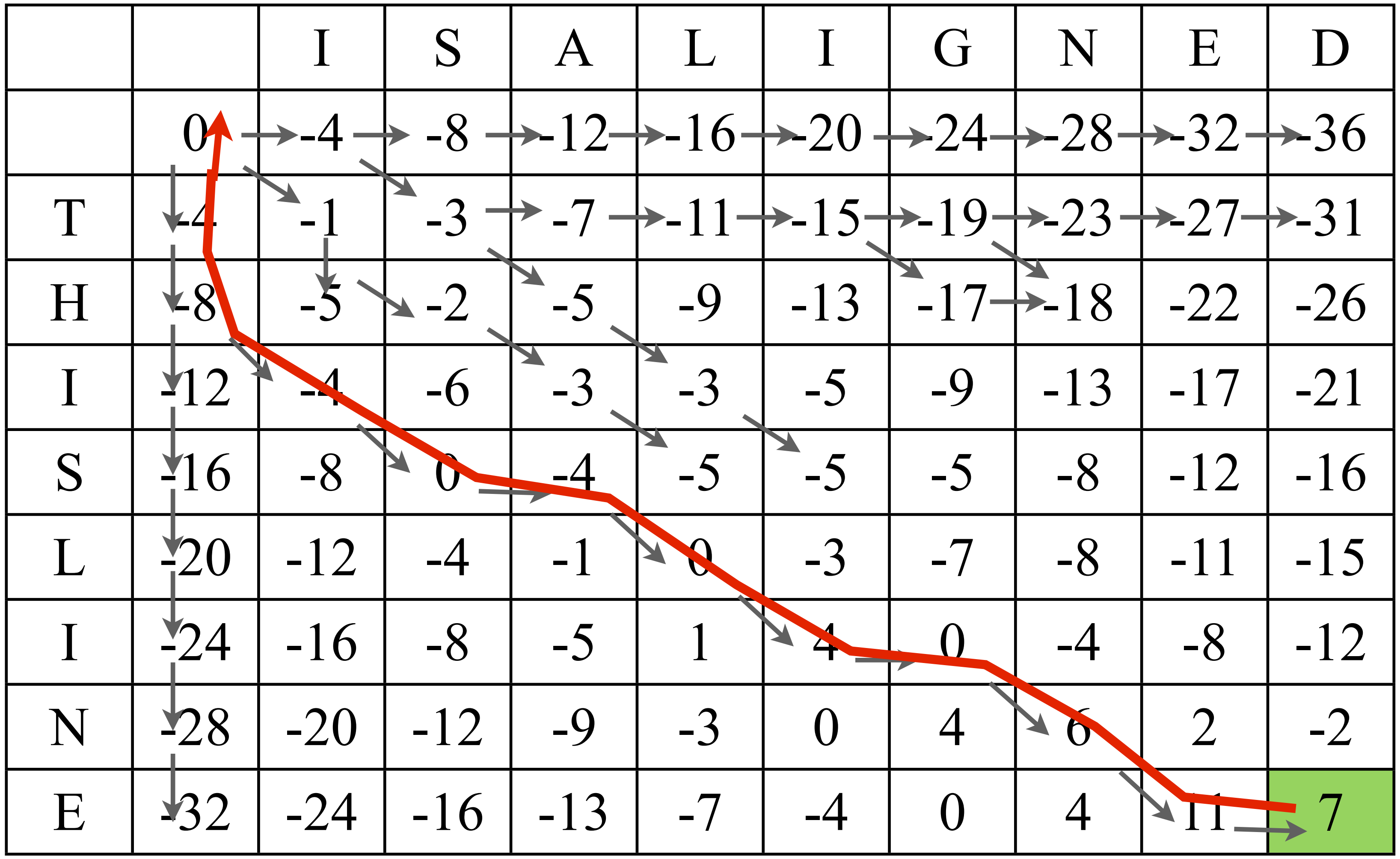
-	-	A	B	C
D	E	F	-	-

Beginning traceback

Needleman-Wunsch Algorithm (Cont.)

Example

Optimal global alignment of V = THISLINE and W= ISALIGNED with gap = -4n_{gap} , score matrix = BLOSUM62



		I	S	A	L	I	G	N	E	D
	0	-4	-8	-12	-16	-20	-24	-28	-32	-36
T	-4	-1	-3	-7	-11	-15	-19	-23	-27	-31
H	-8	-5	-2	-5	-9	-13	-17	-18	-22	-26
I	-12	-4	-6	-3	-3	-5	-9	-13	-17	-21
S	-16	-8	0	-4	-5	-5	-5	-8	-12	-16
L	-20	-12	-4	-1	0	-3	-7	-8	-11	-15
I	-24	-16	-8	-5	1	4	0	-4	-8	-12
N	-28	-20	-12	-9	-3	0	4	6	2	-2
E	-32	-24	-16	-13	-7	-4	0	4	11	7

From: Understanding Bioinformatics by Zvelebil, Baum

V:	T	H	I	S	-	L	I	-	N	E	-
W:	-	-	I	S	A	L	I	G	N	E	D

Needleman-Wunsch Algorithm Efficiency

Runtime: O(nm)

Why?

Space: O(nm)

Needleman-Wunsch Algorithm

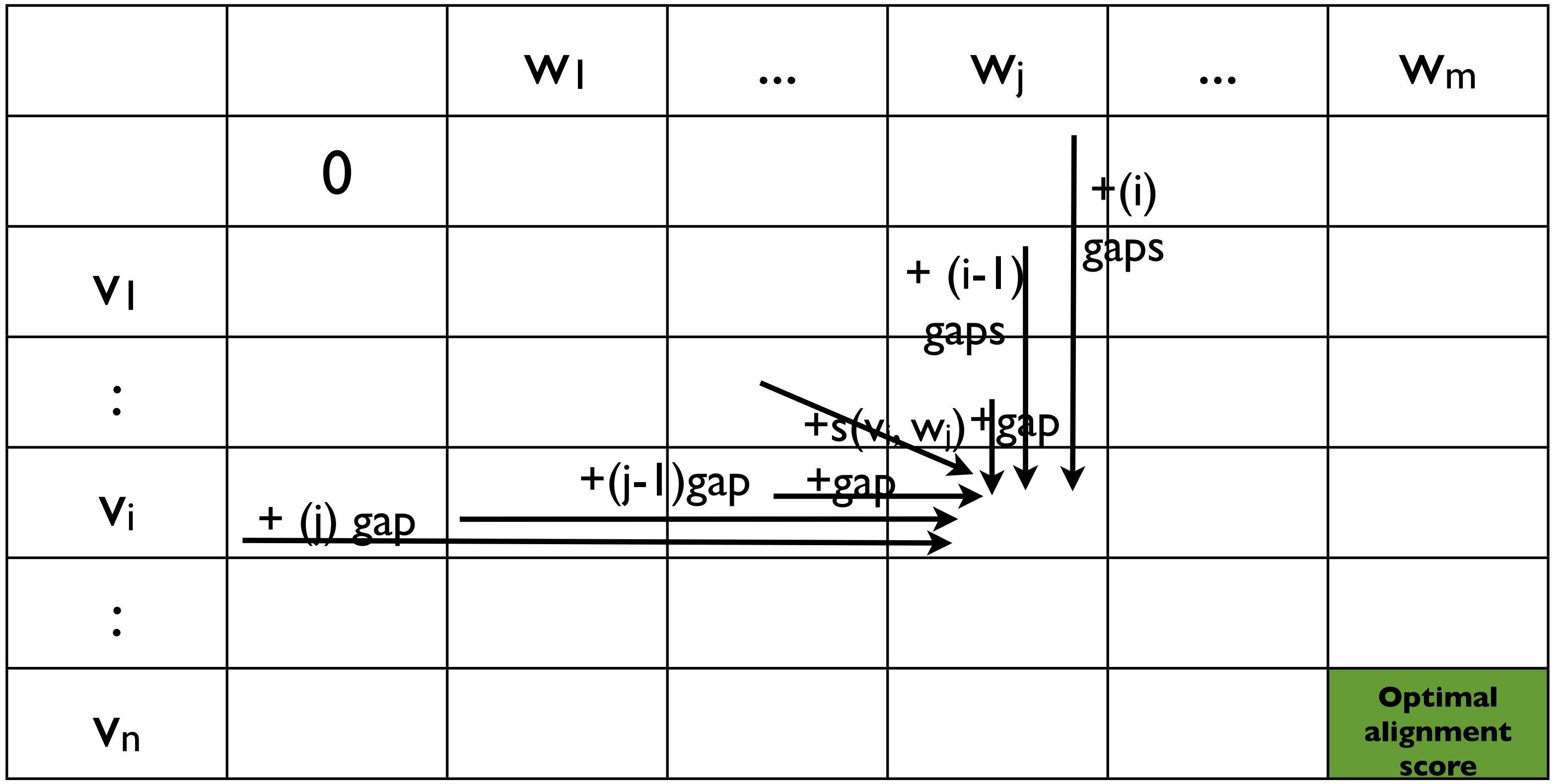
for any gap penalty models

Does affine gap model work with simple Needleman-Wunsch Algorithm we just saw?

Needleman-Wunsch Algorithm

for any gap penalty models

Does affine gap model work with simple Needleman-Wunsch Algorithm we just saw?

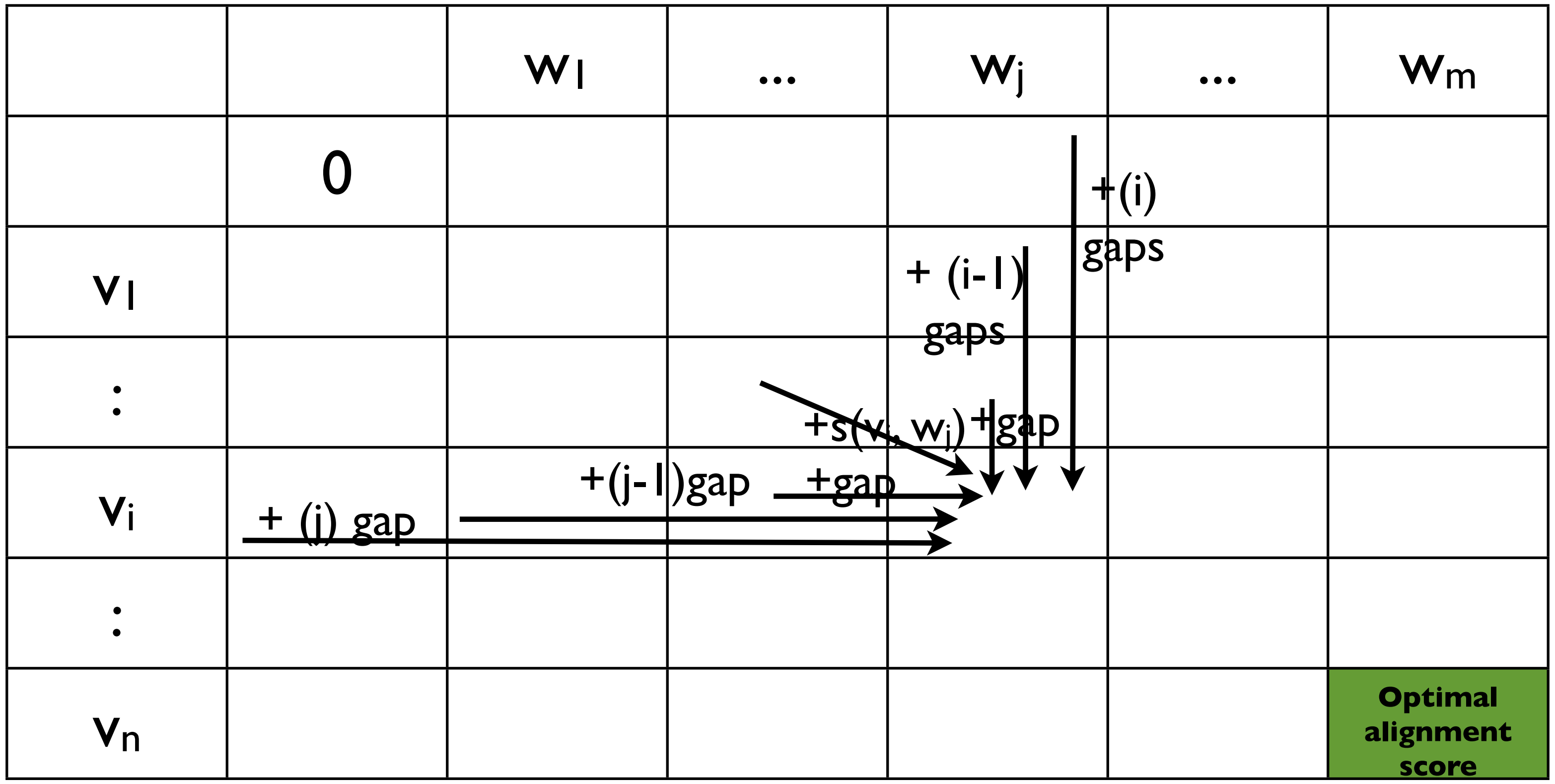


		w_1	…	w_j	…	w_m
	0					
v_1						
:						
v_i						
:						
v_n						Optimal alignment score

Needleman-Wunsch Algorithm

for any gap penalty models

Does affine gap model work with simple Needleman-Wunsch Algorithm we just saw?



		w1	…	wj	…	wm
	0					
v1						
:						
vi						
:						
vn						Optimal alignment score

+(i) gaps

+ (i-1) gaps

+s(vi, wj) +gap

+(j-1)gap

+gap

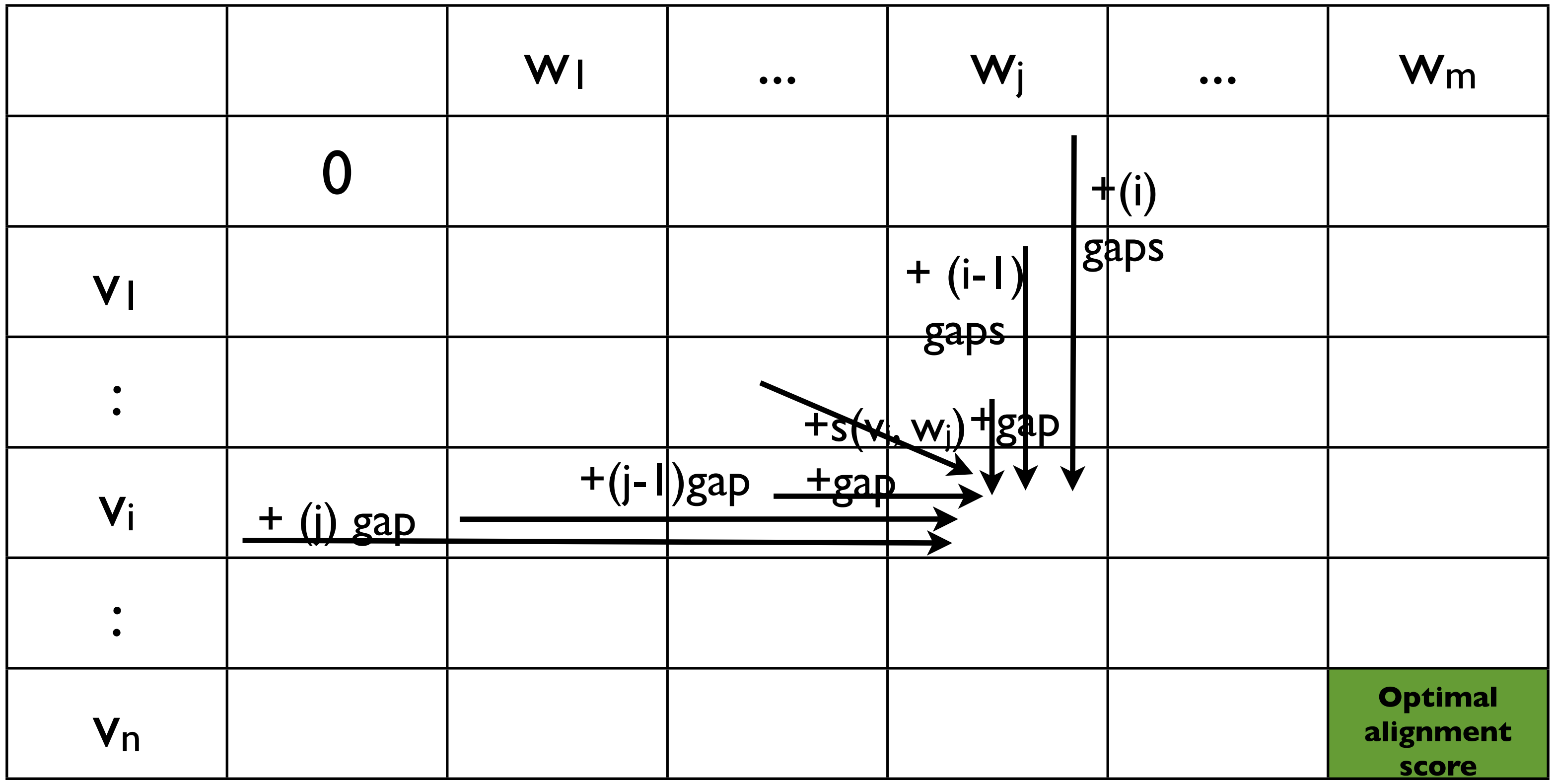
+ (j) gap

$$NW(i,j) = \max \begin{cases} NW(i-1,j-1) + s(v_i, w_j) & \text{match/mismatch} \\ [NW(i-n_{gap}, j) + w(n_{gap1})]_{1 \le n_{gap1} \le i} & \text{delete} \\ [NW(i, j-n_{gap2}) + w(n_{gap2})]_{1 \le n_{gap2} \le j} & \text{insert} \end{cases}$$

Needleman-Wunsch Algorithm

for any gap penalty models

Does affine gap model work with simple Needleman-Wunsch Algorithm we just saw?



		w_1	…	w_j	…	w_m
	0					
v_1						
:						
v_i						
:						
v_n						Optimal alignment score

+(i) gaps

+ (i-1) gaps

+s(v_i, w_j) +gap

+(j-1)gap

+gap

+ (j) gap

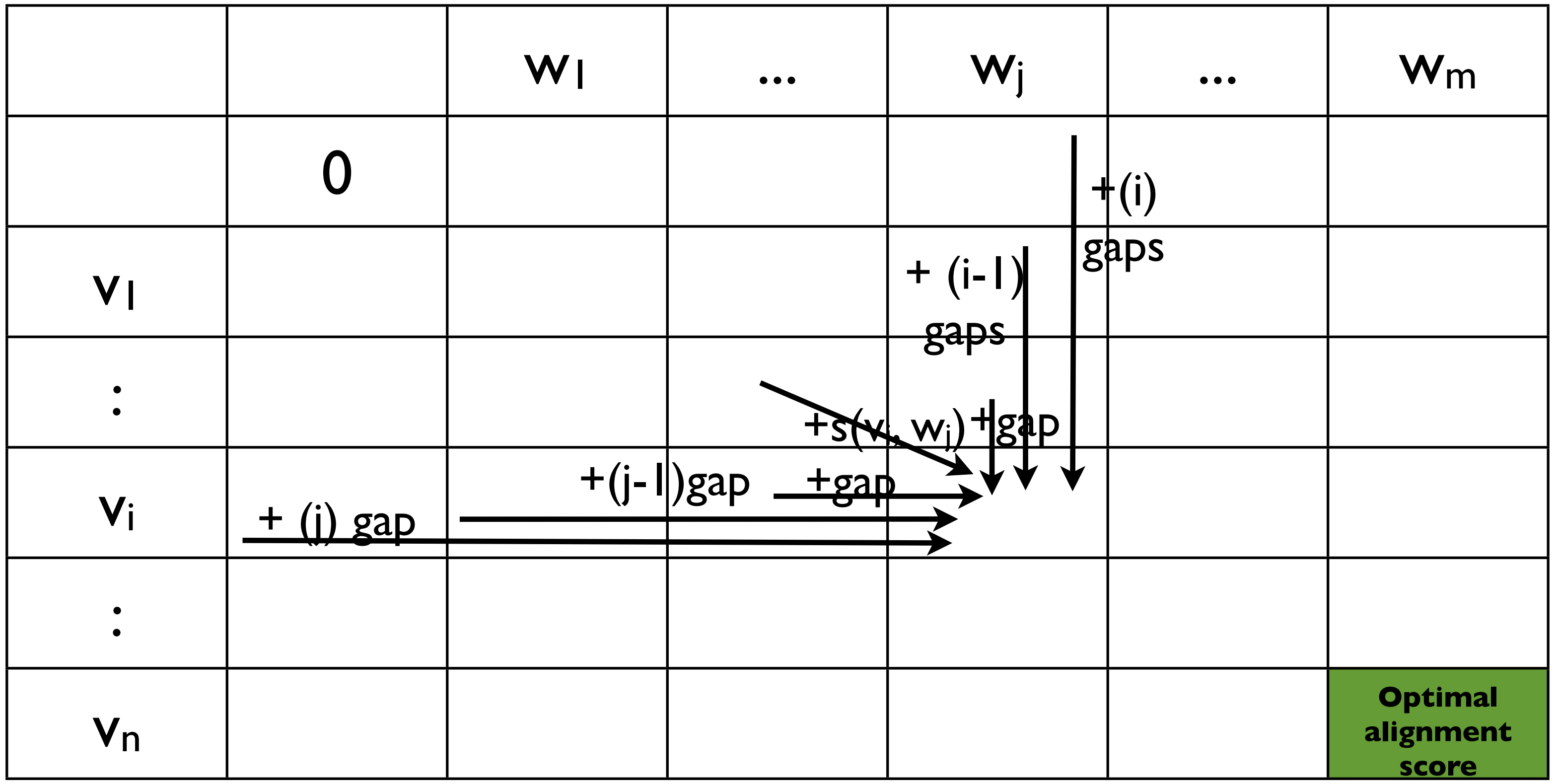
$$NW(i,j) = \max \begin{cases} NW(i-1,j-1) + s(v_i, w_j) & \text{match/mismatch} \\ [NW(i-n_{gap}, j) + w(n_{gap1})]_{1 \le n_{gap1} \le i} & \text{delete} \\ [NW(i, j-n_{gap2}) + w(n_{gap2})]_{1 \le n_{gap2} \le j} & \text{insert} \end{cases}$$

What is the runtime? or space?

Needleman-Wunsch Algorithm

for any gap penalty models

Does affine gap model work with simple Needleman-Wunsch Algorithm we just saw?



		w_1	…	w_j	…	w_m
	0					
v_1						
:						
v_i						
:						
v_n						Optimal alignment score

$$NW(i,j) = \max \begin{cases} NW(i-1, j-1) + s(v_i, w_j) & \text{match/mismatch} \\ [NW(i-n_{gap}, j) + w(n_{gap1})]_{1 \le n_{gap1} \le i} & \text{delete} \\ [NW(i, j-n_{gap2}) + w(n_{gap2})]_{1 \le n_{gap2} \le j} & \text{insert} \end{cases}$$

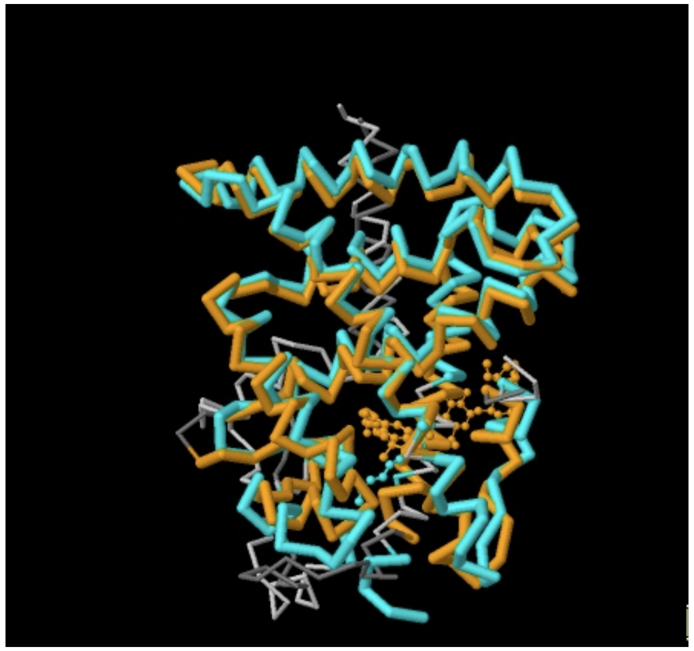
What is the runtime? or space? O(mn^2) where n>m

Local Alignment

finds the most similar regions of a nucleotide or amino acid sequence ignoring other segments of the sequences

Local alignment programs are useful for detecting shared domains in multi-domain proteins.

1FCZ: A T K C I I K I V E F A K R L P G F T G L S I A A C L D I L M L R I C
3U9Q: S V E A V Q E I T E Y A K S I P G F V N L D L N D Q V T L L K Y G V H



1FCZ and 3U9Q superimposed

Optimal local alignments are produced using Smith-Waterman Algorithm

Smith-Waterman Algorithm

A dynamic programming algorithm for optimal local alignment

Given:

Two sequences $V = (v_1 v_2 ... v_n)$ and $W = (w_1 w_2 ... w_m)$.

$(|V| = n$ and $|W| = m)$

Goal:

Find the highest scoring alignment for best subsequence match. The score is usually a measure of similarity.

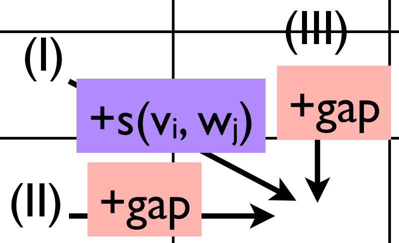
Requirement:

- A matrix SW of optimal scores of subsequence alignments. SW has size $(n+1)x(m+1)$.
- Score matrix
- Defined gap penalty

Smith-Waterman Algorithm (Cont.)

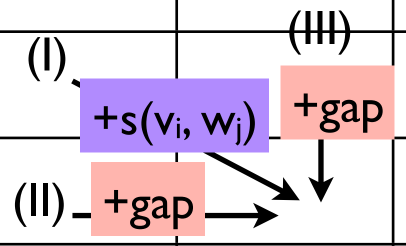
Smith-Waterman Algorithm (Cont.)

		w_1	…	w_j	…	w_m
	0	0	0	0	0	0
v_1	0					
:	0					
v_i	0					
:	0					
v_n	0					



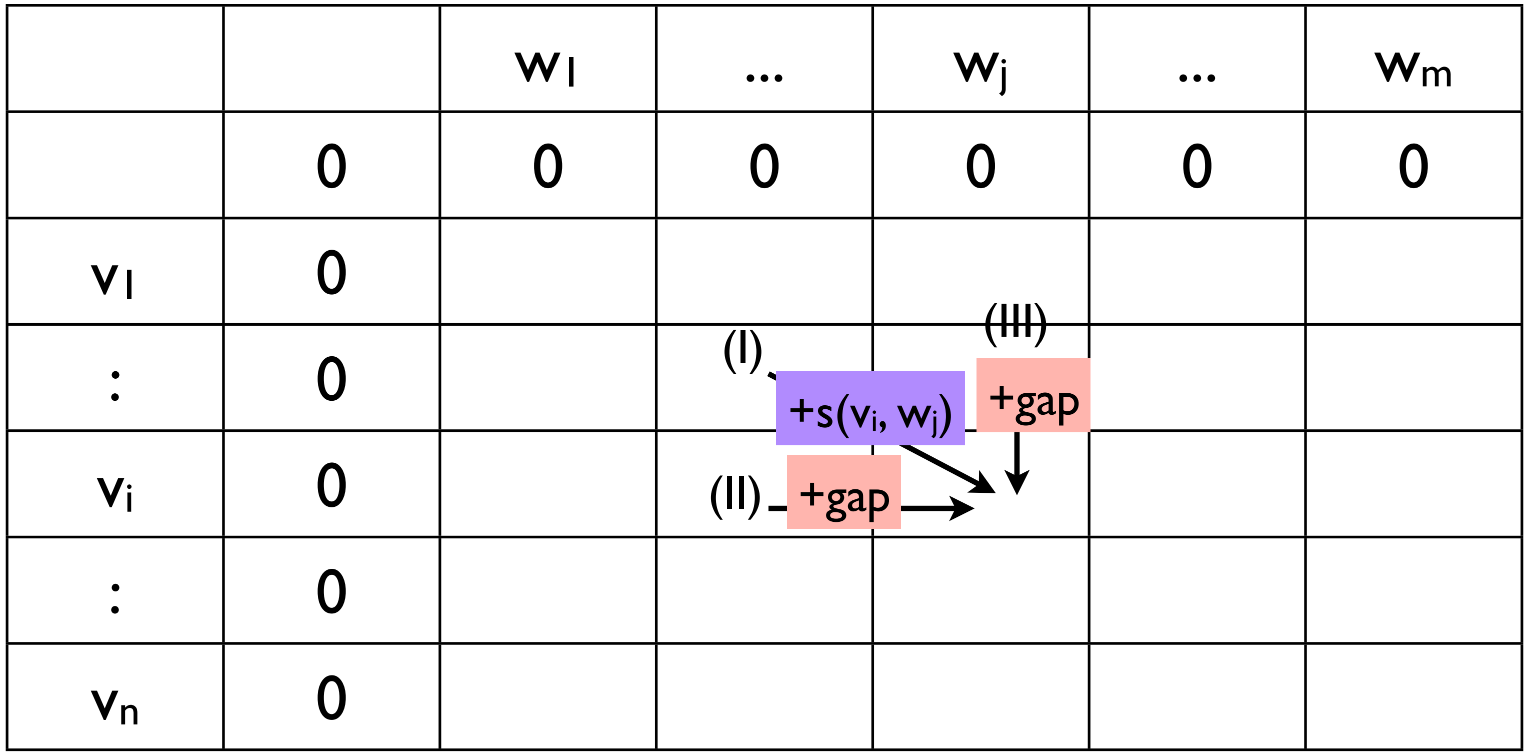
Smith-Waterman Algorithm (Cont.)

		w_1	…	w_j	…	w_m
	0	0	0	0	0	0
v_1	0					
:	0					
v_i	0					
:	0					
v_n	0					



Optimal alignment score = $\max_{0\le i\le n, 0\le j\le m}\{SW(i,j)\}$

Smith-Waterman Algorithm (Cont.)



		w_1	…	w_j	…	w_m
	0	0	0	0	0	0
v_1	0					
:	0					
v_i	0					
:	0					
v_n	0					

Optimal alignment score = $\max_{0 \le i \le n, 0 \le j \le m} \{SW(i,j)\}$

For linear gap penalty model

Base case: SW(i,j) = 0 where i= 0 or j=0

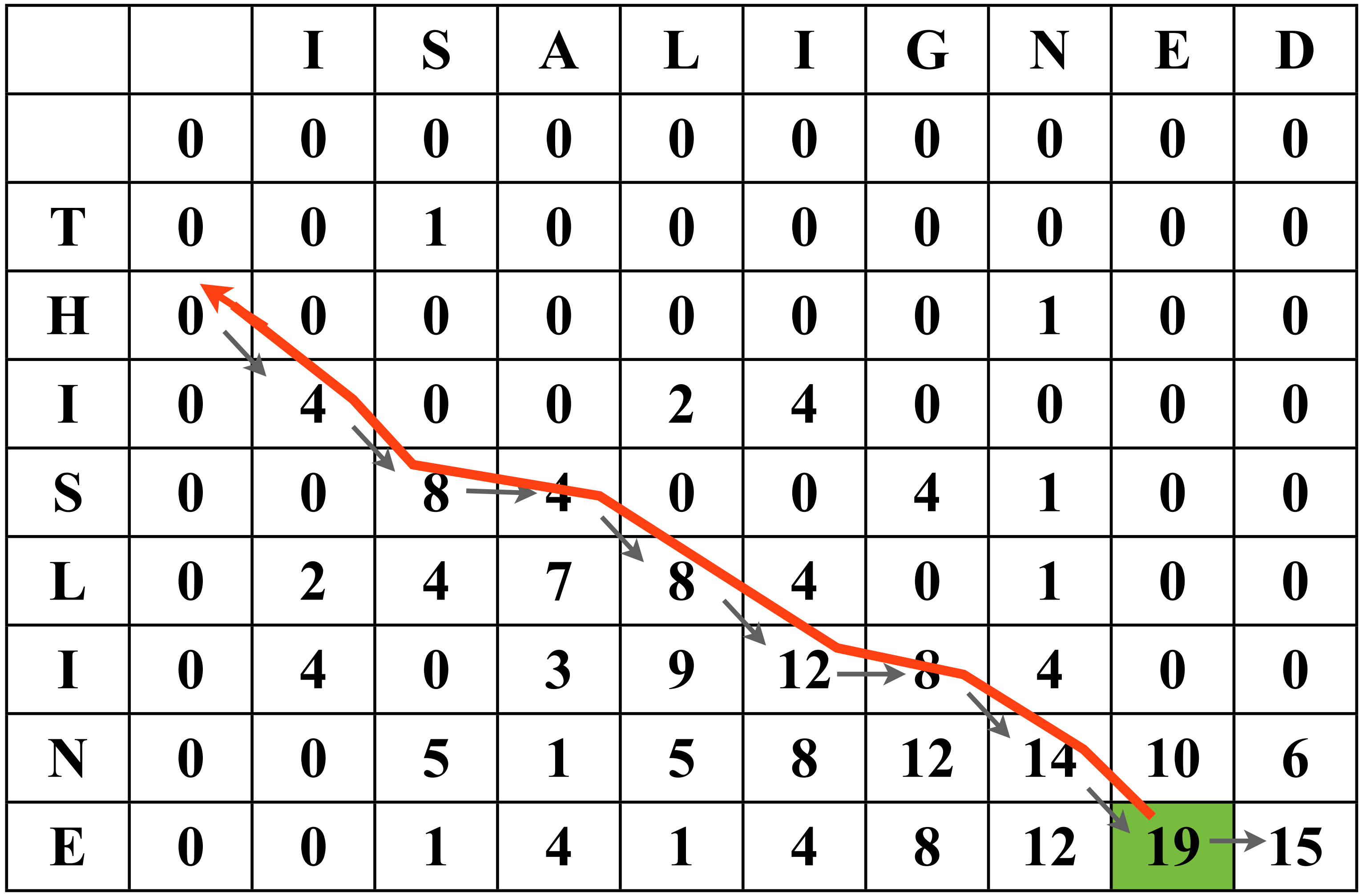
Recurrence:

$$SW(i,j) = \max \begin{cases} 0 & \text{align empty strings} \\ SW(i-1,j-1) + s(v_i, w_j) & \text{match/mismatch} \\ SW(i-1,j) + g & \text{delete} \\ SW(i,j-1) + g & \text{insert} \end{cases}$$

Smith-Waterman Algorithm (Cont.)

Example

Local alignment of V = THISLINE , W= ISALIGNED with gap = -4n_{gap} , score matrix = BLOSUM62



		I	S	A	L	I	G	N	E	D
	0	0	0	0	0	0	0	0	0	0
T	0	0	1	0	0	0	0	0	0	0
H	0	0	0	0	0	0	0	1	0	0
I	0	4	0	0	2	4	0	0	0	0
S	0	0	8	4	0	0	4	1	0	0
L	0	2	4	7	8	4	0	1	0	0
I	0	4	0	3	9	12	8	4	0	0
N	0	0	5	1	5	8	12	14	10	6
E	0	0	1	4	1	4	8	12	19	15

From: Understanding Bioinformatics by Zvelebil, Baum

Traceback starts at entry containing the optimal alignment score.

V:	I	S	-	L	I	-	N	E
W:	I	S	A	L	I	G	N	E

Smith-Waterman Algorithm Efficiency

Runtime: O(nm)

Why?

Space: O(nm)

Which alignment to use?

Example 1. Overlap detection: Aligning exon to a gene sequence

V = ATCCGAACATCCAATCGAAGC   W = AGCATGCAAT

Aligning scores: match = 2, gap = -1 mismatch = -2

Which alignment to use?

Example 1. Overlap detection: Aligning exon to a gene sequence

V = ATCCGAACATCCAATCGAAGC W = AGCATGCAAT

Aligning scores: match = 2, gap = -1 mismatch = -2

Global alignment?

Which alignment to use?

Example 1. Overlap detection: Aligning exon to a gene sequence

V = ATCCGAACATCCAATCGAAGC   W = AGCATGCAAT

Aligning scores: match = 2, gap = -1 mismatch = -2

Global alignment?

```
A T C C G A A C A T C C A A T C G A A G C
A - - - G - - C A T G C A A T - - - - - -
```

Score = 2(9) -1(11) -2(1) = 5

Which alignment to use?

Example 1. Overlap detection: Aligning exon to a gene sequence

V = ATCCGAACATCCAATCGAAGC   W = AGCATGCAAT

Aligning scores: match = 2, gap = -1 mismatch = -2

Global alignment?

```
A T C C G A A C A T C C A A T C G A A G C
A - - - G - - C A T G C A A T - - - - - -
```

Score = 2(9) -1(11) -2(1) = 5

Local alignment?

Which alignment to use?

Example 1. Overlap detection: Aligning exon to a gene sequence

V = ATCCGAACATCCAATCGAAGC   W = AGCATGCAAT

Aligning scores: match = 2, gap = -1 mismatch = -2

Global alignment?

```
A T C C G A A C A T C C A A T C G A A G C
A - - - G - - C A T G C A A T - - - - - -
```

Score = 2(9) -1(11) -2(1) = 5

Local alignment?

```
C A T C C A A T
C A T G C A A T
```

Score = 2(7) -2(1) = 12

Which alignment to use?

Example 1. Overlap detection: Aligning exon to a gene sequence

V = ATCCGAACATCCAATCGAAGC W = AGCATGCAAT

Aligning scores: match = 2, gap = -1 mismatch = -2

Global alignment?

```
A T C C G A A C A T C C A A T C G A A G C
A - - - G - - C A T G C A A T - - - - - -
```

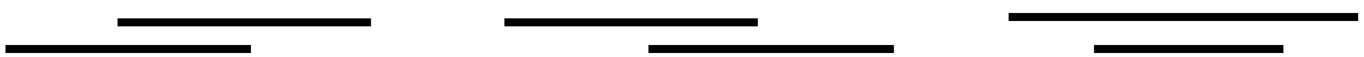
Score = 2(9) -1(11) -2(1) = 5

Local alignment?

```
C A T C C A A T
C A T G C A A T
```

Score = 2(7) -2(1) = 12

Where was the overlap exactly?



Which alignment to use?

example 1 continued

What if avoid penalizing the gaps at the beginning and /or the end of an alignment?

```
A T C C G A - C A T C C A A T C G A A G C
- - - - - A G C A T G C A A T - - - - - -
```

Score = 2(8) -1(1) -2(1) = 13

Spaces in front or end of the exon might be UTR, introns, or enhancer and etc. Thus these gaps should not be penalized.

Which alignment to use?

example 1 continued

What if avoid penalizing the gaps at the beginning and /or the end of an alignment?

```
A T C C G A - C A T C C A A T C G A A G C
- - - - - A G C A T G C A A T - - - - - -
```

Score = 2(8) -1(1) -2(1) = 13

Spaces in front or end of the exon might be UTR, introns, or enhancer and etc. Thus these gaps should not be penalized.

Semi-global alignment. Globally aligning the two sequence but ignoring penalizing gaps at both ends of a sequence.

Which alignment to use? (Cont.)

Example 2. Overlap detection: Sequence assembly:

V = ACCTCACGATCCGA W = TCAACGATCACCGCA

```
---------ACCTCACGATCCGA
TCAACGATCACCGCA--------
```

Semi-global alignment. Globally aligning the two sequence but ignoring penalizing the starting gaps of a sequence and the trailing gaps of the other sequence.

Semi-Global Alignment

finds optimal alignment without penalizing gaps on the ends of the alignment

How to perform semi-global alignment?

Modify the basic Needleman-Wunsch algorithm:

Set the first row and first column of the DP matrix to 0.

		w_1	…	w_j	…	w_m
	0	0	0	0	0	0
v_1	0					
:	0		(I) $+s(v_i, w_j)$	(III) +gap		
v_i	0		(II) +gap			
:	0					
v_n	0					

Optimal alignment score = max (row_n, column_m)

Traceback starts at entry containing the optimal alignment score and ends at the first row or the first column.

Versatility of DP Algorithm

- Memory usage can be optimized
- Runtime can be improved

Versatility of DP Algorithm

- Memory usage can be optimized
- Runtime can be improved
- Heuristically can improve the runtime:
  - FASTA
  - BLAST

References

- Gusfield D. Algorithms on strings, trees and sequences. The press syndicate of the University of Cambridge;1997. p.215-244.

- Zvelebil M, Baum JO. Understanding Bioinformatics. New York: Garland Science; 2008. p. 126-137.

- Steipe B. Homology I: Principles. BCH441 Lecture Fall 2010.

- Jones NC, Pevzner AP. An Introduction to Bioinformatics Algorithms. London: The MIT press; 2004. p. 177-181

- Wing-kin S. Algorithms in Bioinformatics: A practical introduction. London: CRC Press; 2010. p. 32-42.

- Setubal J, Meidanis J. Introduction to Computational Molecular Biology. London: The MIT press. 2007. Chapter 3.

Any Question?