Course organization

- **Introduction (Week 1-2)**
  - Course introduction
  - A brief introduction to molecular biology
  - A brief introduction to sequence comparison
- **Part I: Algorithms for Sequence Analysis (Week 3 - 8)**
  - Chapter 1-3, Models and theories
    - Probability theory and Statistics (Week 3)
    - Algorithm complexity analysis (Week 4)
    - Classic algorithms (Week 5)
  - Chapter 4. Sequence alignment (week 6)
  - Chapter 5. Hidden Markov Models (week 7）
  - Chapter 6. Multiple sequence alignment (week 8)
- **Part II: Algorithms for Network Biology (Week 9 - 16)**
  - Chapter 7. Omics landscape (week 9)
  - Chapter 8. Microarrays, Clustering and Classification (week 10)
  - Chapter 9. Computational Interpretation of Proteomics (week 11)
  - Chapter 10. Network and Pathways (week 12,13)
  - Chapter 11. Introduction to Bayesian Analysis (week 14,15)
  - Chapter 12. Bayesian networks (week 16)

SHANGHAI JIAO TONG UNIVERSITY

Chapter 3: Dynamic Programming （动态编程）

Chaochun Wei

Spring 2018

Contents

- Reading materials
- Introduction
  - Dynamic programming （动态编程）
  - Greedy algorithm （贪心算法）

Reading

Cormen book:

Thomas, H. ,Cormen, Charles, E., Leiserson, and Ronald, L., Rivest . Introduction to Algorithms, The MIT Press.

(read Chapter 16 and 17, page 299-355).

Dynamic programming

- **Find an optimal solution to a problem**
- **Four steps to develop a dynamic programming algorithm**
  1. **Characterize the structure of an optimal solution**
  2. **Recursive formula for an optimal solution**
  3. **Compute the value of an optimal solution**
  4. **Construct an optimal solution from the computed information**

Elements of dynamic programming

- **Two elements are required**
  1. **Optimal substructure （最优子结构）**
     - An optimal solution contains within it optimal solutions to the subproblems
  2. **Overlapping subproblems （重叠子问题）**
     - Recursive formula exists

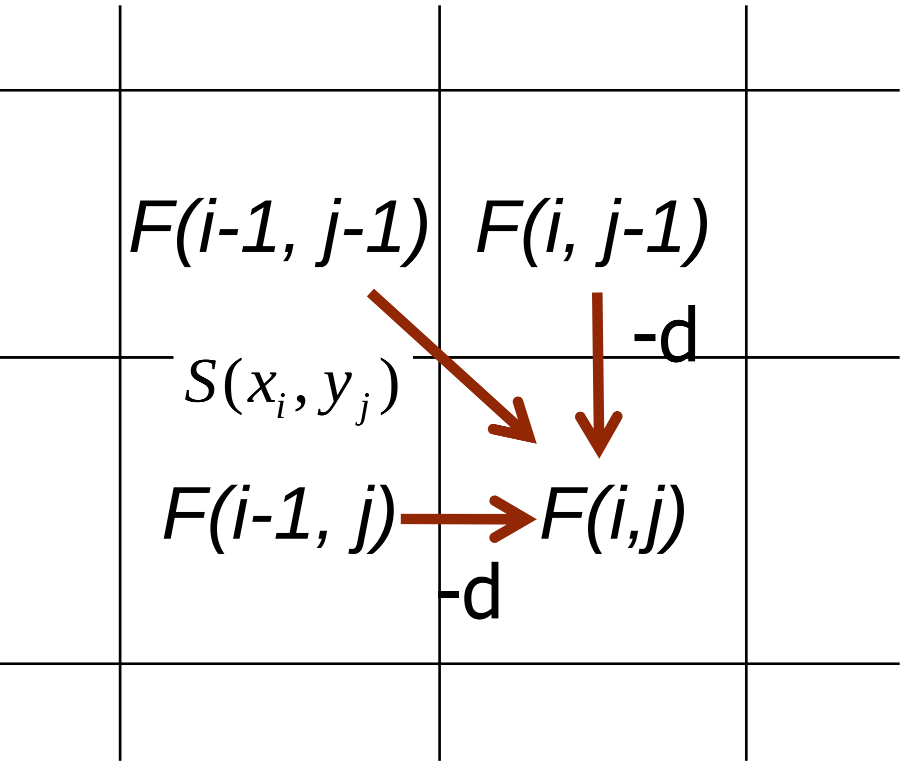
Needleman/Wunsch global alignment (1970)

- Two sequences $X = x_1 \ldots x_n$ and $Y = y_1 \ldots y_m$
- Let $F(i, j)$ be the optimal alignment score *of* $X_{1\ldots i}$ of X up to x_i and $Y_{1\ldots j}$ of Y up to y_j $(0 \leq i \leq n,\ 0 \leq j \leq m)$, *then we have*

$$F(0,0) = 0$$

$$F(i,j) = \max \begin{cases} F(i-1, j-1) + s(x_i, y_j) \\ F(i-1, j) - d \\ F(i, j-1) - d \end{cases}$$

Needleman/Wunsch global alignment (1970)

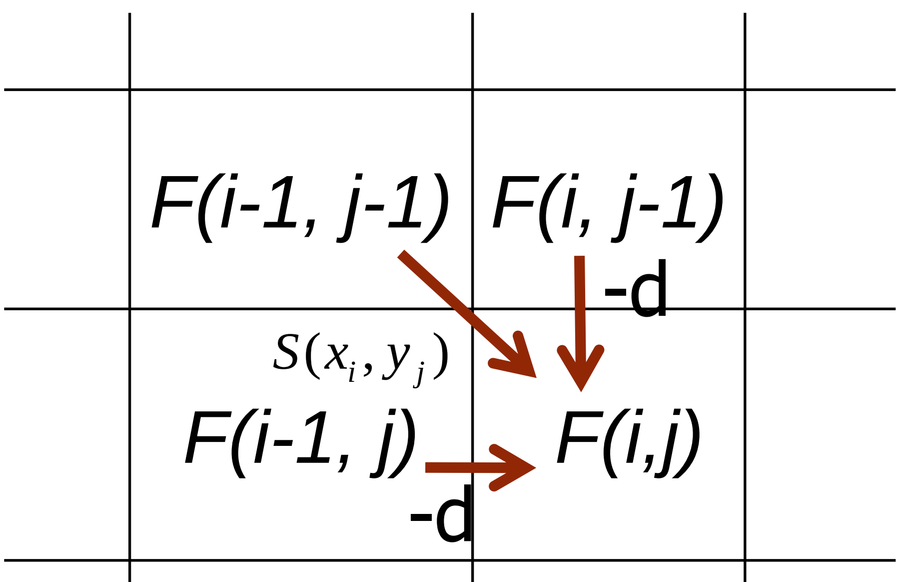


$$F(0,0) = 0$$

$$F(i,j) = \max \begin{cases} F(i-1, j-1) + s(x_i, y_j) \\ F(i-1, j) - d \\ F(i, j-1) - d \end{cases}$$

Elements of dynamic programming

- **Two elements are required**
  1. **Optimal substructure （最优子结构）**
     - An optimal solution contains within it optimal solutions to the subproblems
  2. **Overlapping subproblems （重叠子问题）**
     - Recursive formula exists



$$F(0,0)=0$$

$$F(i,j)=\max\begin{cases} F(i-1,j-1)+s(x_i,y_j) \\ F(i-1,j)-d \\ F(i,j-1)-d \end{cases}$$

Optimal substructure (最优子结构） of Needleman/Wunsch global alignment

- Two sequences $X = x_1 \ldots x_n$ and $Y = y_1 \ldots y_m$
- Let $F(i, j)$ be the optimal alignment score *of* $X_{1\ldots i}$ of X up to X_i and $Y_{1\ldots j}$ of Y up to y_j $(0 \le i \le n,\ 0 \le j \le m)$, *then we have*
  - *1.* if x_i is aligned to y_j then F(i-1,j-1) is the optimal alignment score of $X_{1,\ldots,i-1}$ of X up to x_{i-1} and $Y_{1,\ldots,j-1}$ of Y up to y_{j-1}.
  - 2. if x_i is aligned to no base in Y , then F(i-1,j) is the optimal alignment score of $X_{1,\ldots,i-1}$ of X up to x_{i-1} and $Y_{1,\ldots,j}$ of Y up to y_j.
  - 3. if y_j is aligned to no base in X , then F(i,j-1) is the optimal alignment score of $X_{1,\ldots,i}$ of X up to x_i and $Y_{1,\ldots,j-1}$ of Y up to y_{j-1}.

Optimal substructure of Needleman/Wunsch global alignment

- Two sequences $X = x_1 \ldots x_n$ and $Y = y_1 \ldots y_m$
- Let $F(i, j)$ be the optimal alignment score *of* $X_{1\ldots i}$ of X up to X_i and $Y_{1\ldots j}$ of Y up to Y_j $(0 \le i \le n,\ 0 \le j \le m)$

  - 1. if x_i is aligned to y_j then F(i-1,j-1) is the optimal alignment score of $X_{1,\ldots,i-1}$ of X up to x_{i-1} and $Y_{1,\ldots,j-1}$ of Y up to y_{j-1}.

    Proof If x_i is aligned to y_j, and F(i-1,j-1) is not the optimal alignment score of $X_{1,\ldots,i-1}$ and $Y_{1,\ldots,j-1}$, then there is an optimal alignment of $X_{1,\ldots,i-1}$ and $Y_{1,\ldots,j-1}$ with a score F′(i-1,j-1) higher than F(i-1,j-1). Then the alignment of F′(i-1,j-1) adding the alignment of x_i and y_j has a higher score than F(i,j), which is a contradiction.

Optimal substructure of Needleman/Wunsch global alignment

- Two sequences $X = x_1 \ldots x_n$ and $Y = y_1 \ldots y_m$
- Let $F(i, j)$ be the optimal alignment score *of* $X_{1\ldots i}$ of X up to X_i and $Y_{1\ldots j}$ of Y up to Y_j $(0 \le i \le n, 0 \le j \le m)$

  - 2. if X_i is aligned to no base in Y , then F(i-1,j) is the optimal alignment score of $X_{1,\ldots,i-1}$ of X up to X_{i-1} and $Y_{1,\ldots,j}$ of Y up to Y_j.
  - **Proof** If X_i is aligned to no base in Y, and F(i-1,j) is not the optimal alignment score of $X_{1,\ldots,i-1}$ and $Y_{1,\ldots,j}$, then there is an optimal alignment of $X_{1,\ldots,i-1}$ and $Y_{1,\ldots,j}$ with a score F′(i-1,j) higher than F(i-1,j). Then the alignment of F′(i-1,j) adding the alignment of X_i and *a gap* has a higher score than F(i,j), which is a contradiction.

Optimal substructure of Needleman/Wunsch global alignment

- Two sequences $X = x_1 \ldots x_n$ and $Y = y_1 \ldots y_m$
- Let $F(i, j)$ be the optimal alignment score *of* $X_{1\ldots i}$ of X up to X_i and $Y_{1\ldots j}$ of Y up to Y_j $(0 \le i \le n,\ 0 \le j \le m)$

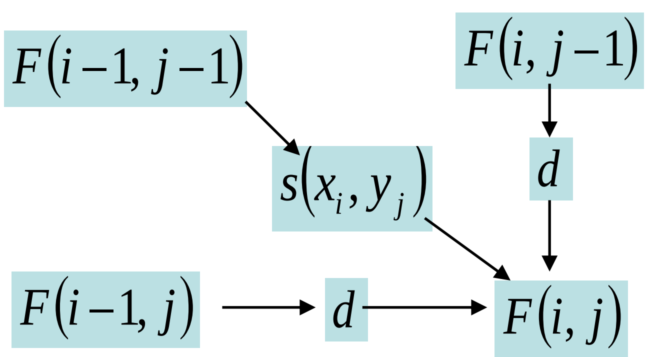
  - 3. if Y_j is aligned to no base in X , then F(i,j-1) is the optimal alignment score of $X_{1\ldots,i}$ of X up to X_i and $Y_{1\ldots,j-1}$ of Y up to Y_{j-1}.
  - **Proof** is symmetric to 2.

A simple example

	A	C	G	T
A	2	-7	-5	-7
C	-7	2	-7	-5
G	-5	-7	2	-7
T	-7	-5	-7	2

Find the optimal alignment of AAG and AGC.
Use a gap penalty of d=-5.

		A	A	G
A				
G				
C				

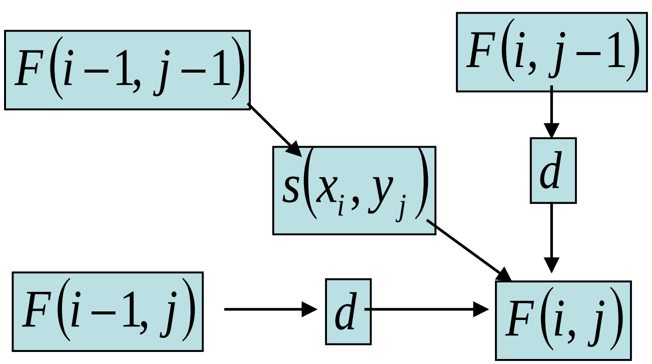


A simple example

	A	C	G	T
A	2	-7	-5	-7
C	-7	2	-7	-5
G	-5	-7	2	-7
T	-7	-5	-7	2

Find the optimal alignment of AAG and AGC.
Use a gap penalty of d=-5.

		A	A	G
	0			
A				
G				
C				

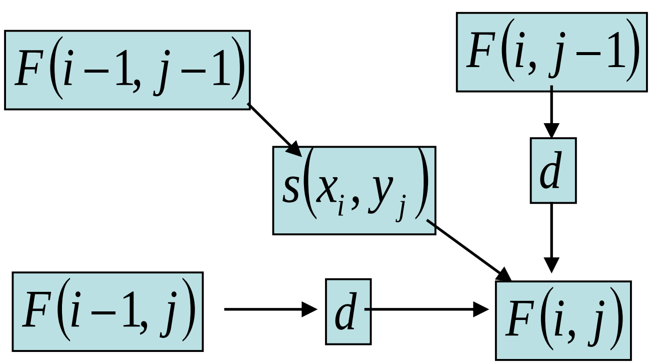


A simple example

	A	C	G	T
A	2	-7	-5	-7
C	-7	2	-7	-5
G	-5	-7	2	-7
T	-7	-5	-7	2

Find the optimal alignment of AAG and AGC.
Use a gap penalty of d=-5.

		A	A	G
	0	-5	-10	-15
A	-5			
G	-10			
C	-15			

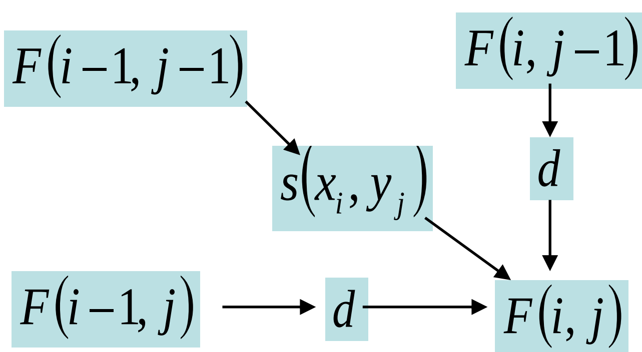


A simple example

	A	C	G	T
A	2	-7	-5	-7
C	-7	2	-7	-5
G	-5	-7	2	-7
T	-7	-5	-7	2

Find the optimal alignment of AAG and AGC.
Use a gap penalty of d=-5.

		A	A	G
	0	-5	-10	-15
A	-5	2	-3	-8
G	-10	-3	-3	-1
C	-15	-8	-8	**-6**



Traceback

1. **Start from the lower right corner and trace back to the upper left.**
2. **Each arrow introduces one character at the end of each aligned sequence.**
3. **A <u>horizontal</u> move puts a gap in the <u>left</u> sequence.**
4. **A <u>vertical</u> move puts a gap in the <u>top</u> sequence.**
5. **A diagonal move uses one character from each sequence.**

A simple example

1. Start from the lower right corner and trace back to the upper left.
2. Each arrow introduces one character at the end of each aligned sequence.
3. A horizontal move puts a gap in the left sequence.
4. A vertical move puts a gap in the top sequence.
5. A diagonal move uses one character from each sequence.

Find the optimal alignment of AAG and AGC.
Use a gap penalty of d=-5.

		A	A	G
	0	-5		
A		2	-3	
G				-1
C				**-6**

```
AAG-        AAG-
-AGC        A-GC
```

Exercise

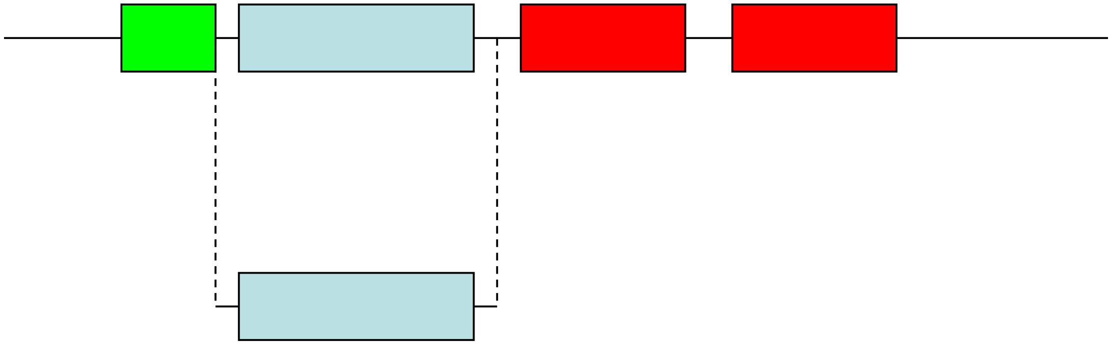
- Find Global alignment
  - X=catgt
  - Y=acgctg
  - Score: d=-1 mismatch=-1 match=2

Answer

i \ j		0	1	2	3	4	5
i			c	a	t	g	t
0		0	-1	-2	-3	-4	-5
1	a	-1	-1	1	0	-1	-2
2	c	-2	1	0	0	-1	-2
3	g	-3	0	0	-1	2	1
4	c	-4	-1	-1	-1	1	1
5	t	-5	-2	-2	1	0	3
6	g	-6	-3	-3	0	3	2

X (row c a t g t); Y (column a c g c t g)

Local alignment



- A single-domain protein may be homologous to a region within a multi-domain protein.
- Usually, an alignment that spans the complete length of both sequences is not required.

Smith/Waterman local alignment (1981)

- Two sequences $X = x_1 \ldots x_n$ and $Y = y_1 \ldots y_m$
- Let $F(i, j)$ be the optimal alignment score *of* $X_{1\ldots i}$ of X up to x_i and $Y_{1\ldots j}$ of Y up to Y_j $(0 \le i \le n,\ 0 \le j \le m)$, *then we have*

$$F(0,0) = 0$$

$$F(i,j) = \max\begin{cases} 0 \\ F(i-1, j-1) + s(x_i, y_j) \\ F(i-1, j) - d \\ F(i, j-1) - d \end{cases}$$

Local alignment

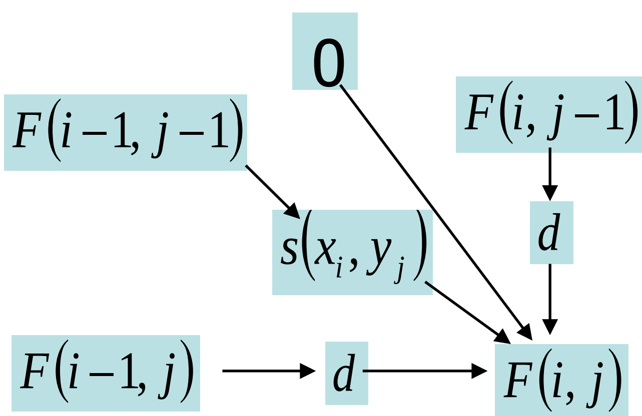
- **Two differences with respect to global alignment:**
  - **No score is negative.**
  - **Traceback begins at the highest score in the matrix and continues until you reach 0.**
- **Global alignment algorithm: *Needleman-Wunsch*.**
- **Local alignment algorithm: *Smith-Waterman*.**

A simple example

	A	C	G	T
A	2	-7	-5	-7
C	-7	2	-7	-5
G	-5	-7	2	-7
T	-7	-5	-7	2

Find the optimal local alignment of AAG and AGC.
Use a gap penalty of d=-5.

		A	A	G
A				
G				
C				

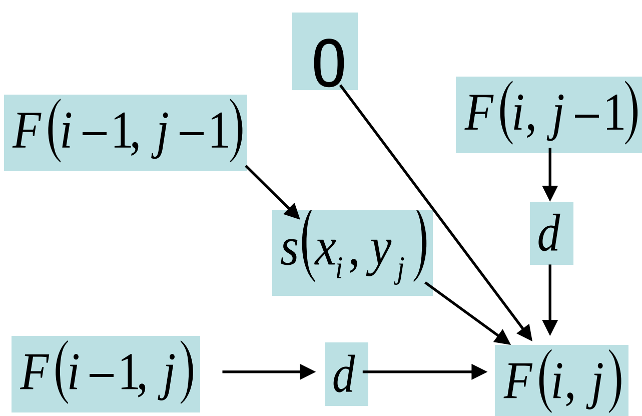


A simple example

	A	C	G	T
A	2	-7	-5	-7
C	-7	2	-7	-5
G	-5	-7	2	-7
T	-7	-5	-7	2

Find the optimal local alignment of AAG and AGC.
Use a gap penalty of d=-5.

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

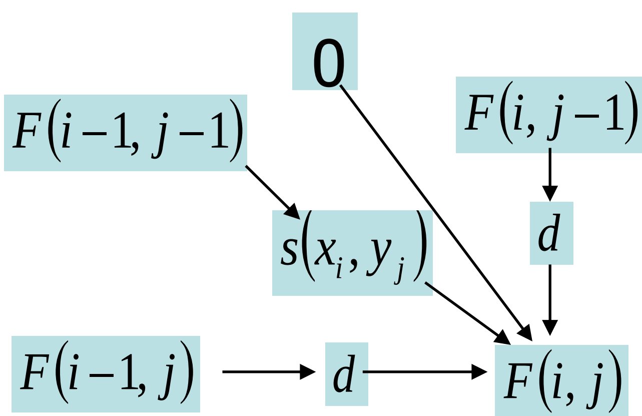


A simple example

	A	C	G	T
A	2	-7	-5	-7
C	-7	2	-7	-5
G	-5	-7	2	-7
T	-7	-5	-7	2

Find the optimal local alignment of AAG and AGC.
Use a gap penalty of d=-5.

		A	A	G
	0	0	0	0
A	0	2	2	0
G	0	0	0	4
C	0	0	0	0

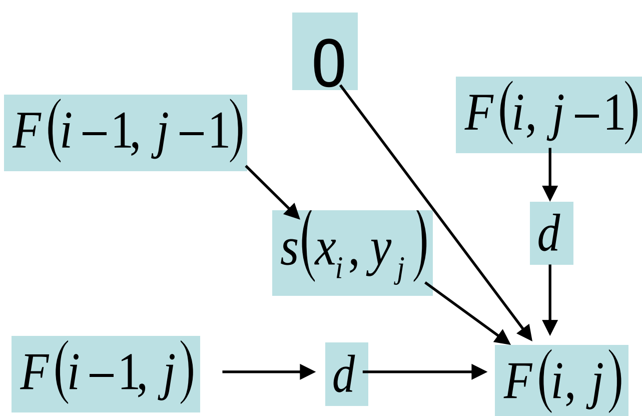


A simple example

	A	C	G	T
A	2	-7	-5	-7
C	-7	2	-7	-5
G	-5	-7	2	-7
T	-7	-5	-7	2

Find the optimal local alignment of AAG and AGC.
Use a gap penalty of d=-5.

		A	A	G
	0	0	0	0
A	0	2	2	0
G	0	0	0	4
C	0	0	0	0



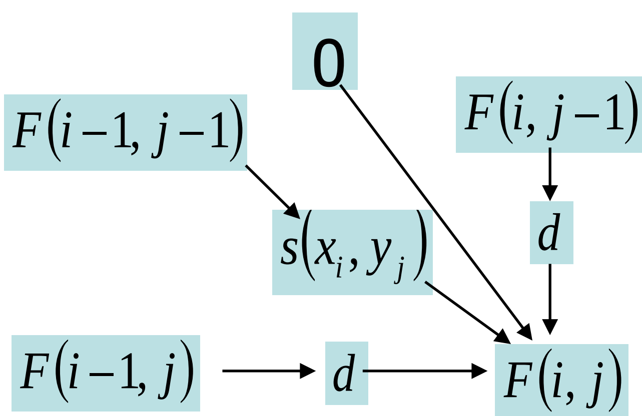
AG
AG

Local alignment

	A	C	G	T
A	2	-7	-5	-7
C	-7	2	-7	-5
G	-5	-7	2	-7
T	-7	-5	-7	2

Find the optimal local alignment of AAG and GAAGGC.
Use a gap penalty of d=-5.

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

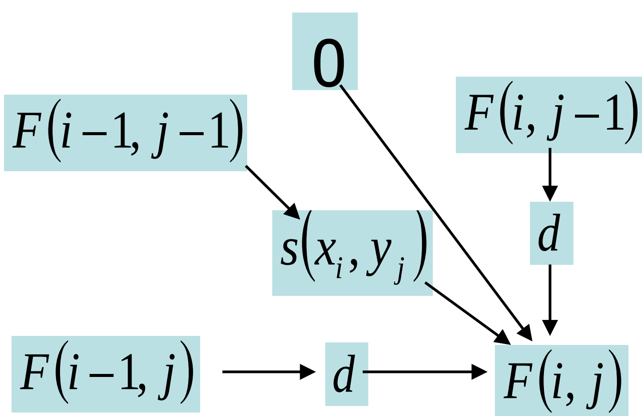


Local alignment

	A	C	G	T
A	2	-7	-5	-7
C	-7	2	-7	-5
G	-5	-7	2	-7
T	-7	-5	-7	2

Find the optimal local alignment of AAG and GAAGGC.
Use a gap penalty of d=-5.

		A	A	G
	0	0	0	0
G	0	0	0	2
A	0	2	2	0
A	0	2	4	0
G	0	0	0	6
G	0	0	0	2
C	0	0	0	0



Greedy algorithm（贪心算法）: Choose the best at the moment

- **Not always produce the optimal result**
- **Two elements are required to find an optimal solution by greedy algorithm**

1. **Greedy-choice property**
   - Global optimal can be reached by local optimal (greedy)
2. **Optimal substructure**
   - An optimal solution contains within it optimal solutions to the subproblems

Greedy Algorithm

Example: Activity-selection problem

- **N activities: S={1, 2, …, N}. Only one can be selected at a time. Select the maximum number of mutually compatible activities**

Let s_i and f_i be the start time and finish time for activity i.
Assume $s_1 \leq s_2 \leq ... \leq s_N$

```
GREEDY_ACTIVITY_SELECTION(s,f)
N ← length[s]
A← {1}
j← 1
for i ← 2 to N
        do if  s_i>= f_j
              then A← A U {i}
                    j← i
Return A
```

Greedy Algorithm

Example: Activity-selection problem

- **N activities: S={1, 2, …, N}. Only one can be selected at a time. Select the maximum number of mutually compatible activities**

Let s_i and f_i be the start time and finish time for activity i.

Assume $f_1 \le f_2 \le \ldots \le f_N$

```
GREEDY_ACTIVITY_SELECTION(s,f)
N ← length[s]
A← {1}
j← 1
for i ← 2 to N
        do if  s_i >= f_j
              then A← A U {i}
                        j← i
Return A
```

Proof the solution is optimal!

Acknowledgement

PPTs for examples in dynamic programming are kindly provided by Dr. Qi Liu.